

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Índice de Anexos

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES

ELABORACIÓN DE TARJETAS DE CIRCUITO IMPRESO

APÉNDICE B

CÓDIGO DE PROGRAMACIÓN DEL MICROCONTROLADOR SILABS

ANEXO A

UNIDAD DE TRANSMISIÓN INALÁMBRICA PARA ESTACIONES AUTOMATIZADAS DE ADQUISICIÓN DE DATOS DE TRÁFICO AUTOMOTOR. (APÉNDICES Y ANEXOS)

LAYOUT DE LAS TABLAS DE CIRCUITO IMPRESO

APÉNDICE ESTRUCTURA DE LOS MÓDULOS DE LA UDAT

PRUEBA DEL FUNCIONAMIENTO GENERAL DE LA EAAT

APÉNDICE F

SECUENCIA DE COMANDOS AT ENTRE LA UDCAT Y EL MÓDULO

ANYDATA EMU-800

ANEXO A

DATASHEET DEL AUTONSENSE II

ANEXO B

DATASHEET DEL MÓDULO ANYDATA EMU-800

ANEXO C

REALIZADO POR

DATASHIRT DEL MICROCONTROLADOR

Ángel Alberto Bracho Biondo.
Raúl Enrique Carpio Domínguez.

PROFESOR GUÍA

Ing. Iñaki Mendizábal.

FECHA

Caracas, 13 de octubre de 2006.

Índice de Anexos

APÉNDICE A	68
ELABORACIÓN DE TARJETAS DE CIRCUITO IMPRESO	68
APÉNDICE B	71
CÓDIGO DE PROGRAMACIÓN DEL MICROCONTROLADOR SILABS C8051F120.....	71
APÉNDICE C	94
GLOSARIO DE ACRÓNIMOS.....	94
APÉNDICE D	99
LAYOUT DE LAS TARJETAS DE CIRCUITO IMPRESO	99
APÉNDICE E	102
PRUEBA DEL FUNCIONAMIENTO GENERAL DE LA EAADTA	102
APÉNDICE F	106
SECUENCIA DE COMANDOS AT ENTRE LA UDCAT Y EL MÓDULO ANYDATA EMII-800.....	106
ANEXO A.....	108
DATASHEET DEL AUTOSENSE II	108
ANEXO B.....	136
DATASHEET DEL MÓDULO ANYDATA EMII-800.....	136
ANEXO C	156
DATASHORT DEL MICROCONTROLADOR SILABS C8051F120	156

Con el fin de obtener el hardware que conforma la unidad de radio en modo de radio se debe realizar el siguiente proceso y obtener los resultados previos que se indican a continuación:

- 1.- Utilizando para transfer se imprime el diagrama circular para luego ser plasmado sobre una lámina de cobre con la intención de plasmar las trazas en la tarjeta, tal y como se muestra en la Figura A.1. Posteriormente, y con la ayuda de un taladro, se abren los agujeros que servirán más adelante para introducir los dispositivos electrónicos que conforman el hardware.

Apéndice A

Elaboración de tarjetas de circuito impreso

Figura A.1. Vista de la tarjeta posterior al proceso de

- 2.- Una vez obtenido el diagrama sobre la tarjeta, ésta se coloca dentro de un baño que acelera la corrosión del cobre (Ver Figura A.2) y permanece que solamente queden las líneas conductoras deseadas.

Con el fin de obtener el hardware que conforma la unidad de almacenamiento, se debe realizar el siguiente proceso y obtener los resultados previos que se listan a continuación:

- 1.- Utilizando papel transfer se imprime el diagrama circuital para luego ser planchado sobre una lámina de cobre con la intención de plasmar las líneas en dicha tarjeta, tal y como se muestra en la Figura A.1. Posteriormente, y con la ayuda de un taladro, se abren los agujeros que servirán más adelante para introducir los dispositivos electrónicos que conforman el hardware.

Figura A.1. Visión de la tarjeta durante el proceso de corrosión.

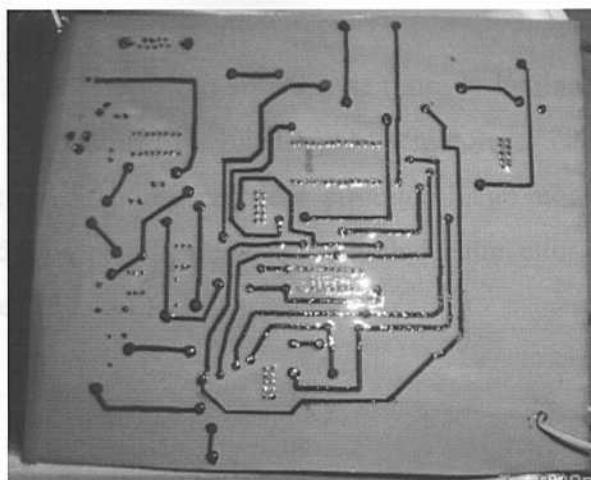


Figura A.1. Visión de la tarjeta posterior al planchado

- 2.- Una vez obtenido el diagrama sobre la tarjeta, ésta se sumerge dentro de un ácido que acelera la corrosión del cobre (Ver Figura A.2) y permite que solamente queden impresas las líneas conductoras deseadas.

Figura A.2. Visión de la tarjeta finalizada.

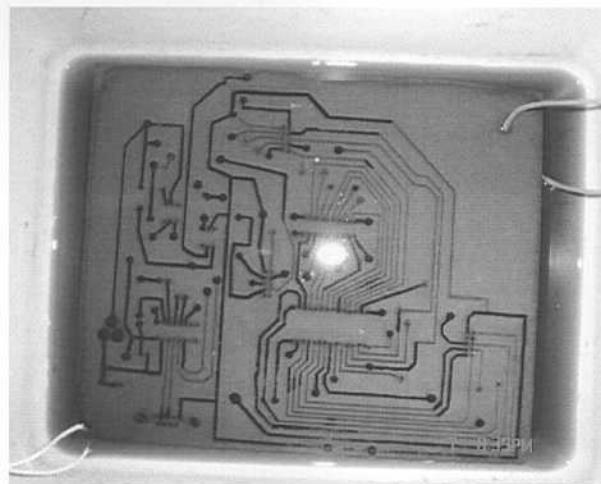


Figura A.2. Visión de la tarjeta durante el proceso de corrosión.

3.- Luego de alcanzarse el objetivo anterior, se retira la lámina del ácido, se remoja en agua limpia y se deja secar. El siguiente paso será colocar cables en los huecos de paso por la condición de doble cara del circuito, luego ubicar cada uno de los componentes y soldarlos para lograr conductividad entre ellos, quedando la tarjeta como el resultado de la Figura A.3.

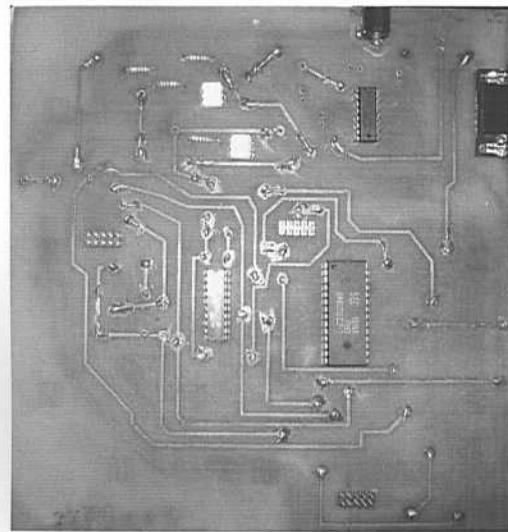


Figura A.3. Visión de la tarjeta finalizada.

```
#include <8051F120.h>
#include <util.h>
#include "stack.h"
#include "math.h"

// Punto de conexión UAPL
// #include <uart.h>

// Punto de conexión TELNET
// #include <telnet.h>

// Punto de conexión MLL
// #include <mll.h>
```

Apéndice B

Código de Programación del microcontrolador SILABS C8051F120

```
#include "SYNCLK.h"
void PORT_init(void);
void USART_init(void);
void USART_send(unsigned char);
void USART_recieve(unsigned char *el_caracter_recibido);
void UART_init(void);
void UART_rx_isr(void);
void USART1_IRQHandler(void);
```

```
//-----  
// Includes  
//-----  
  
#include <c8051f120.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
  
//-----  
// CONSTANTES Globales  
//-----  
  
#define BAUDRATE    115200      // Baud rate of UART in bps  
#define INTCLK     24500000      // Internal oscillator frequency en Hz  
#define SYSCLK     49000000      // Salida del PLL (INTCLK*2)  
sbit at 0x96 LED='1';          // green LED: '1' = ON; '0' = OFF  
sbit at 0x97 DISC='0';         // Pin para la desconexion del modem  
sbit at 0xDA A10='1';  
sbit at 0xD9 A9='1';  
sbit at 0xD8 A8='1';  
sbit at 0xEF A7='1';  
sbit at 0xEE A6='1';  
sbit at 0xED A5='1';  
sbit at 0xEC A4='1';  
sbit at 0xEB A3='1';  
sbit at 0xEA A2='0';  
sbit at 0xE9 A1='0';  
sbit at 0xE8 A0='0';  
sbit at 0xDB WRITE='1';  
sbit at 0xDD ENABLE='0';  
sbit at 0xDC G='0';  
sbit at 0xF8 D0='0';  
sbit at 0xF9 D1='0';  
sbit at 0xFA D2='0';  
sbit at 0xFB D3='0';  
sbit at 0xFC D4='0';  
sbit at 0xFD D5='0';  
sbit at 0xFE D6='0';  
sbit at 0xFF D7='0';  
  
//-----  
// Funciones  
//-----  
  
void SYSCLK_Init (void);  
void PORT_Init (void);  
void UART_Init (void);  
void EMI_Init(void);  
void Interrupts_Init(void);  
int chequeo (int cont,int vel,int auxi);  
void Limpiar_tipo(void);  
void Limpiar_transmitir(void);  
void UART1_ISR (void) interrupt 20;
```

```
void Timer3_ISR (void) interrupt 14; // ISR del timer3
void UART0_ISR (void) interrupt 4;
void wait_ms (int ms);
void Limpiar_cont(void); // variable que indica si hay que enviar los datos
void Limpiear_tot(void);
void Limpiear_vel(void); // en cuanto se cambien la secuencia de envío
void Vueltas_tot(void);
void Recopilacion_datos(void);
void ARRANCAR_RELOJ(void);
void LEER_HORA(void); // para leer la hora del reloj
void Limpiar_vtemp(void);
//-----
// VARIABLES Globales
//-----
unsigned char xdata malloc_mempool [0x1000];
int i,int p=0;
int xdata minutos=0; int xdata hora=0; int xdata dia=0; int xdata mes=0; int xdata ano=0;
int xdata tipo[12];
int xdata vel[12];
int xdata tot[12];
int xdata cont[12];
int xdata receptor[7000];int xdata transmitir[34];int xdata vtemp[12]; int xdata vpromedio=0;
long tim5min=0; int xdata hola=0; long tim1hora=0;int datos;int xdata acum2=0;
int acum=0;int xdata b;int xdata k=0;int mandar=0; int resp[30]; int h=0;
int xdata s=0; long xdata a=0; int xdata estado=1;int xdata intento=0;int xdata d=1;int xdata c=1;

//-----
// MAIN Routine
//-----
void main (void) {
    WDTCN = 0xde; // deshabilitar watchdog timer
    WDTCN = 0xad;
    SYSCLK_Init(); // inicializacion del oscilador
    PORT_Init(); // inicializacion de los puertos
    UART_Init(); // inicializacion UART1
    EMI_Init(); // inicializacion memoria externa
    Interrupts_Init(); // inicializacion interrupciones
    Limpiear_cont(); // Limpieza de arreglos importantes
    Limpiear_tot();
    Limpiear_vel();
    Limpiear_tipo();
    Limpiear_transmitir();
    Limpiear_vtemp();
    ARRANCAR_RELOJ(); // Para inicializar el oscilador del reloj
    externo
    EA = 1; // Habilitar las interrupciones globalmente
    resp[0]=0;

    receptor[0]=0;
    DISC=!DISC;

    while (1) {
```

```
LED = !LED; // cambio del estado del LED

if(mandar==1)//bandera "mandar" que indica si hay que enviar los datos
{
    LED ='1'; // LED encendido cuando se establezca la secuencia de envio
    tim1hora=0;
    tim5min=0;

    LEER_HORA();           //Para leer la hora del reloj externo
    Limpiar_transmitir();
    transmitir[1]=1;//identificacion del equipo
    transmitir[2]=estado;
    transmitir[3]=dia;
    transmitir[4]=mes;
    transmitir[5]=ano;
    transmitir[6]=hora;
    transmitir[7]=minutos;
    b=8;
    s=1;
    while(s<12)
    {
        transmitir[b]=tot[s];
        transmitir[b+1]=vtemp[s];// Se llena el arreglo que se enviara con los valores establecidos
        b=b+2;
        s=s+1;
    }
    transmitir[30]=10;
    Vueltas_tot();
    p=0;
    EA=0;
    SFRPAGE = UART0_PAGE;
    TI0=1;
    SBUF0=0x41;//Secuencia de envio del comando AT+CRM=130
    TI0=0;
    wait_ms(1);
    TI0=1;
    SBUF0=0x54;
    TI0=0;
    wait_ms(1);
    TI0=1;
    SBUF0=0x2B;
    TI0=0;
    wait_ms(1);
    TI0=1;
    SBUF0=0x43;
    TI0=0;
    wait_ms(1);
    TI0=1;
    SBUF0=0x52;
    TI0=0;
    wait_ms(1);
    TI0=1;
    SBUF0=0x4D;
```

```
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x3D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x31;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x33;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x30;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0A;
TI0=0;
wait_ms(1);
SFRPAGE = CONFIG_PAGE;
TI0=0;
EA=1;
wait_ms(100);
if ((resp[0]==0x0D) & (resp[1]==0x0A) & (resp[2]==0x4F) & (resp[3]==0x4B))//respuesta OK
{
p=0;
EA=0;
SFRPAGE = UART0_PAGE;
TI0=1;
SBUF0=0x41;//Secuencia de envio del comando AT*PID="5176360@cantv.net"
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x54;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x2A;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x50;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x49;
TI0=0;
```

```
wait_ms(1);
TI0=1;
SBUF0=0x44;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x3D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x22;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x35;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x31;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x37;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x36;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x33;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x36;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x30;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x40;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x63;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x61;
TI0=0;
wait_ms(1);
TI0=1;
```

```
SBUF0=0x6E;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x74;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x76;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x2E;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x6E;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x65;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x74;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x22;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0A;
TI0=0;
wait_ms(1);
SFRPAGE = CONFIG_PAGE;
TI0=0;
EA=1;
wait_ms(100);
p=0;
if ((resp[0]==0x0D) & (resp[1]==0x0A) & (resp[2]==0x4F) & (resp[3]==0x4B))//resp OK
{
p=0;
EA=0;
SFRPAGE = UART0_PAGE;
TI0=1;
SBUF0=0x41;//Secuencia de envio del comando AT*PPW="23562"
TI0=0;
wait_ms(1);
TI0=1;
```

```
SBUF0=0x54;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x2A;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x50;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x50;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x57;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x3D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x22;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x32;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x33;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x35;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x36;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x32;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x22;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0D;
TI0=0;
```

```
wait_ms(1);
TI0=1;
SBUF0=0x0A;
TI0=0;
wait_ms(1);
SFRPAGE = CONFIG_PAGE;
TI0=0;
EA=1;
wait_ms(100);
p=0;
if ((resp[0]==0x0D) & (resp[1]==0x0A) & (resp[2]==0x4F) & (resp[3]==0x4B))//resp OK
{
p=0;
EA=0;
SFRPAGE = UART0_PAGE;
TI0=1;
SBUF0=0x41;//Secuencia de envio del comando AT+DIP= "200.002.013.048"
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x54;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x2B;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x44;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x49;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x50;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x3D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x22;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x32;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x30;
TI0=0;
```

```
wait_ms(1);
    TI0=1;
    SBUF0=0x30;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x2E;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x30;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x30;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x30;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x32;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x2E;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x30;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x31;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x33;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x2E;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x30;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x34;
    TI0=0;
wait_ms(1);
    TI0=1;
    SBUF0=0x38;
    TI0=0;
wait_ms(1);
    TI0=1;
```

```
SBUF0=0x22;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0A;
TI0=0;
wait_ms(1);
SFRPAGE = CONFIG_PAGE;
TI0=0;
EA=1;
wait_ms(100);
p=0;
if ((resp[0]==0x0D) & (resp[1]==0x0A) & (resp[2]==0x4F) & (resp[3]==0x4B))//ok
{
p=0;
EA=0;
SFRPAGE = UART0_PAGE;
TI0=1;
SBUF0=0x41;//Secuencia de envio del comando AT+DPORT= "5050"
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x54;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x2B;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x44;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x50;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x4F;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x52;// Secuencia de envío del comando AT+T1023
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x54;
TI0=0;
wait_ms(1);
TI0=1;
```

```
SBUF0=0x3D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x22;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x35;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x30;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x35;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x30;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x22;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0A;
TI0=0;
wait_ms(1);
SFRPAGE = CONFIG_PAGE;
TI0=0;
EA=1;
wait_ms(100);
p=0;
if ((resp[0]==0x0D) & (resp[1]==0x0A) & (resp[2]==0x4F) & (resp[3]==0x4B))//ok
{
p=0;
EA=0;
SFRPAGE = UART0_PAGE;
TI0=1;
SBUF0=0x41;// Secuencia de envio del comando ATDT123
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x54;
TI0=0;
wait_ms(1);
TI0=1;
```

```
SBUF0=0x44;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x54;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x31;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x32;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x33;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0D;
TI0=0;
wait_ms(1);
TI0=1;
SBUF0=0x0A;
TI0=0;
wait_ms(1);
SFRPAGE = CONFIG_PAGE;
TI0=0;
EA=1;
wait_ms(15000);

if ((resp[0]==0x43) & (resp[1]==0x41) & (resp[2]==0x4C) & (resp[3]==0x4C) & (resp[4]==0x0D) &
(resp[5]==0x0A) & (resp[6]==0x50) & (resp[7]==0x50) & (resp[8]==0x50) & (resp[9]==0x0D) &
(resp[10]==0x0A) & (resp[11]==0x43) & (resp[12]==0x4F) & (resp[13]==0x4E) & (resp[14]==0x4E)
& (resp[15]==0x45) & (resp[16]==0x43) & (resp[17]==0x54))//CALL PPP CONNECT
{
mandar=0;
Limpiar_tot();

p=0;
SFRPAGE = CONFIG_PAGE;
EA=0;
SFRPAGE = UART0_PAGE;
h=1;
while (h<42)
{
    TI0=1;
    SBUF0=transmitir[h];//Momento en donde se envian los datos hacia el modulo de transmision
    TI0=0;
    wait_ms(100);
    h=h+1;
}
DISC=!DISC;
wait_ms(1000);
```



```
//-----
// Initialization Subroutines
//-----

//-----
// SYSCLK_Init
//-----
//          // Restore SFR page
// This routine initializes the system clock to use the internal oscillator
// at 24.5 MHz multiplied by two using the PLL.
//
void SYSCLK_Init (void)
{
    int i;           // software timer
    char SFRPAGE_SAVE = SFRPAGE;      // Save Current SFR page
    SFRPAGE = CONFIG_PAGE;           // set SFR page
    OSCICN = 0x83;                  // set internal oscillator to run
                                    // at its maximum frequency
    CLKSEL = 0x00;                  // Select the internal osc. as
                                    // the SYSCLK source
    //Turn on the PLL and increase the system clock by a factor of M/N = 2
    SFRPAGE = CONFIG_PAGE;
    PLL0CN = 0x00;                  // Set internal osc. as PLL source
    SFRPAGE = LEGACY_PAGE;
    FLSCL = 0x10;                  // Set FLASH read time for 50MHz clk
                                    // or less
    SFRPAGE = CONFIG_PAGE;
    PLL0CN |= 0x01;                // Enable Power to PLL
    PLL0DIV = 0x01;                // Set Pre-divide value to N (N = 1)
    PLL0FLT = 0x01;                // Set the PLL filter register for
                                    // a reference clock from 19 - 30 MHz
                                    // and an output clock from 45 - 80 MHz
    PLL0MUL = 0x02;                // Multiply SYSCLK by M (M = 2)
    for (i=0; i < 256; i++) ;     // Wait at least 5us
    PLL0CN |= 0x02;                // Enable the PLL
    while (!(PLL0CN & 0x10));     // Wait until PLL frequency is locked
    CLKSEL = 0x02;                  // Select PLL as SYSCLK source
    SFRPAGE = SFRPAGE_SAVE;        // Restore SFR page
}

//-----
// PORT_Init
//-----
//          // Initialize the I/O ports
//Inicializacion de los puertos de entradas y salida
//
void PORT_Init (void)
{
    char SFRPAGE_SAVE = SFRPAGE;      // Save Current SFR page
    SFRPAGE = CONFIG_PAGE;           // set SFR page
    XBR0 = 0x04;                    // Configure port 0
    XBR1 = 0x00;                    // Configure port 1
    XBR2 = 0x46;                    // Enable crossbar and weak pull-up, e indicar que la EMI es en el p 0
                                    // Enable UART1
```

```
char SFRPAGE_SAVE; // Save Current SFR page
P0MDOUT |= 0x05; // Set TX1 pin to push-pull
//P1MDOUT |= 0x40; // Set P1.6(LED) to push-pull
P1MDOUT |= 0xC0; // Set SFRPAGE_SAVE; // Restore SFR page
P5MDOUT |= 0xFF;
P6MDOUT |= 0xFF;
P7MDOUT |= 0xFF; // SFRPAGE; // Save Current SFR page
SFRPAGE = SFRPAGE_SAVE; // Restore SFR page
}
//-----
//-----  
// UART1_Init
//-----  
//
// Configure the UART1 using Timer1, for <baudrate> and 8-N-1.
// Subarchivos de reporte
void UART_Init (void)
{
    char SFRPAGE_SAVE = SFRPAGE; // Save Current SFR page
    SFRPAGE = UART1_PAGE;
    SCON1 = 0x50; // SCON1: mode 0, 8-bit UART, enable RX
    SFRPAGE = UART0_PAGE;
    SCON0 = 0x50;
    SSTA0 = 0x1F;
    SFRPAGE = TIMER01_PAGE;
    TMOD &= ~0xF0;
    TMOD |= 0x20; // TMOD: timer 1, mode 2, 8-bit reload
    CKCON |= 0x01;
    TH1 = 0x96;
    SFRPAGE = TIMER01_PAGE;
    TMOD = 0x20;
    CKCON = 0x01;
    TH1 = 0x96;
    TL1 = TH1; // initialize Timer1
    TR1 = 1; // start Timer1
    SFRPAGE = TMR3_PAGE;
    TMR3CF = 0x08; // document los
    RCAP3L = 0xFF;
    RCAP3H = 0xFF; // Stop Timer 3
    TR3 = 1; // SFRPAGE_SAVE; // Restore SFRPAGE
    SFRPAGE = TMR4_PAGE;
    TMR4CF = 0x08;
    RCAP4L = 0xE5;
    RCAP4H = 0xFF;
    TR4 = 1;
    SFRPAGE = UART1_PAGE;
// TI1 = 1;
// RI1 = 1;
    SFRPAGE = UART0_PAGE;
// TI0 = 1;
// RI0 = 1; // Indicate TX1 ready
    SFRPAGE = SFRPAGE_SAVE; // Restore SFR page
}
void EMI_Init (void)
{
```

```
char SFRPAGE_SAVE = SFRPAGE;      // Save Current SFR page
    SFRPAGE = EMI0_PAGE;
// EMI0CF = 0x07;
    SFRPAGE = SFRPAGE_SAVE;        // Restore SFR page
}
void Interrupts_Init (void)
{char SFRPAGE_SAVE = SFRPAGE;      // Save Current SFR page
    IE     = 0x90;
    IP     = 0x10;
    EIE2   = 0x41;
    EIP2   = 0x40;
SFRPAGE = SFRPAGE_SAVE;          // Restore SFR page
}
//-----
// Subrutinas de soporte
//-----
// wait_ms
//-----
// Esta rutina genera un retraso de tiempo de <ms> millisegundos.
//
void wait_ms(int ms)
{
    char SFRPAGE_SAVE = SFRPAGE;      //
    SFRPAGE = TMR2_PAGE;
    TMR2CN = 0x00;                  // Stop Timer3; Clear TF3;
    TMR2CF = 0x00;                  // use SYSCLK/12 as timebase
RCAP2H = (-(SYSCLK/1000/12)) >> 8;
RCAP2L = (-(SYSCLK/1000/12));    // Timer 2 overflows at 1 kHz
ET2 = 0;                         // Disable Timer 2 interrupts
TR2 = 1;                         // Start Timer 2
while(ms){
    TF2 = 0;                      // wait until timer overflows
    while(!TF2);                  // wait until timer overflows
    ms--;                         // decrement ms
}
TR2 = 0;                         // Stop Timer 2
SFRPAGE = SFRPAGE_SAVE;          // Restore SFRPAGE/
}
//-----
// Interrupcion de UART 1
//-----
void UART1_ISR (void) interrupt 20 //proceso de recepcion
{
if(TI1==1)
{
    TI1=0;
}
else
{
    if(RI1==1)
    { SFRPAGE = UART1_PAGE;
```

```
RI1=0;
SCON1=0x50;
receptor[acum] = SBUF1; // Se acumulan todos los datos recibidos desde el sensor en un arreglo de memoria
acum = acum + 1;
} // cada 10ms se recibe (cont) interrupt 14/interrupcion de timer para comandos de velocidad
} // cada 5 min chequeo si hay valores distintos
} // en las velocidades de cada tipo
} // cada 10ms se recibe (cont) interrupt 14/interrupcion de timer para comandos de velocidad
//-----
// Limpieza de arreglos de memoria importante
//-----
// 
void Limpiar_transmitir(void) //Limpieza del arreglo de transmision
{int xdata w=1;
while(w < 41)
{transmitir[w]=0;
w=w+1;
}
}
void Limpiar_cont(void)//Limpieza del arreglo de conteo de cada tipo de vehiculo
{int xdata t=1;
while(t < 12)
{cont[t]=0;
t=t+1;
}
}
void Limpiar_tipo(void)//Limpieza del arreglo de cada tipo de vehiculo
{int xdata c=1;
while(c < 12)
{tipo[c]=0;
c=c+1;
}
}
void Limpiar_tot(void)//Limpieza del arreglo del total de cada tipo de vehiculo
{int xdata v=1;
while(v < 12)
{tot[v]=0;
v=v+1;
}
}
void Limpiar_vel(void)//Limpieza del arreglo de velocidad de cada tipo de vehiculo
{int xdata y=1;
while(y < 12)
{vel[y]=0;
y=y+1;
}
}
void Limpiar_vtemp(void)//Limpieza del arreglo de velocidad promedio de cada tipo de vehiculo
{int xdata f=1;
while(f < 12)
{vtemp[f]=0;
f=f+1;
}
}
```

```
//-----  
// Interrupcion del Timer 3  
//-----  
//  
void Timer3_ISR (void) interrupt 14//interrupcion de timer3 para establecer que  
//cada 5 min chequee si hay valores distintos  
//en las velocidades de cada tipo  
{ TF3=0;  
tim5min=tim5min+1;  
tim1hora=tim1hora+1;  
if (tim1hora==1636370000) //Condicional que indica en el momento que se cumplan 60 min  
{  
estado=3;  
if (mandar==0)  
{mandar=1;  
tim1hora=0;  
}  
}  
else //void ISR (void) interrupt 4 //Recibe y envía los datos  
{  
if(acum > 6900) //Condicional que indica si hay memoria disponible  
{Recopilacion_datos();//  
mandar=1;//  
estado=2;//  
};//  
else//  
{};//  
  
if (tim5min == 136000000) //Condicional que indica en el momento que se cumplan 5 min  
//if (tim5min==63600000)  
{Recopilacion_datos(); // Proceso en donde se recopilan los datos y luego se examina si hubo cambio  
de velocidad  
tim5min=0;  
estado=1;  
};//  
}{//  
};//  
};//  
};//  
// Rutina para chequear si hubo cambio de velocidad  
//-----  
//  
int chequeo(int cont1,int vell1,int auxi) // Se compara la velocidad promedio obtenida en el tiempo  
//examinado  
// y se compara con la almacenada previamente, si hay una diferencia  
// de 5 MPH se levanta la bandera "mandar" para establecer la transmision  
{  
if(auxi == 0)  
{  
cont1=0;  
vell1=0;//  
if (mandar==0)//  
{mandar=1;//  
};//
```

```
//  
return vell;  
}  
else  
{if (abs((auxi/(cont1))-vell)>4)  
{vell=(auxi/(cont1));  
if (mandar==0)  
{mandar=1;  
}  
}  
}  
return vell;  
}  
}  
  
//-----  
// Interrupcion de UART0  
//-----  
  
void UART0_ISR (void) interrupt 4 //Recibe y envia los datos hacia el modulo  
{  
    if (TI0==1)  
    {  
        TI0=0;  
    }  
    else  
    {  
        if (RI0==1)  
        {RI0=0;  
        resp[p]=SBUF0;  
        p=p+1;  
        SCON0=0x50;  
        }  
    }  
}  
//-----  
// Recoleccion de los datos  
//-----  
void Recopilacion_datos(void) //Proceso de analisis de los datos recabados en el tiempo establecido  
{  
    //se analizan por tipo para luego realizar la rutina de chequeo  
    SFRPAGE = CONFIG_PAGE;  
    EA=0;  
    a=0;  
    c=1;  
    while(c<16)//  
    {if(receptor[c]==165)//  
    {a=c-1;//  
    break;//  
    }/  
    c=c+1//  
    }/  
    while(a<(acum))  
    {if ((acum-a)>14)//  
    {}/  
    if (receptor[a+4]==1)
```

```
{  
    cont[1] = cont[1]+1;  
    tipo[1] = tipo[1] + receptor[a+14];  
}  
else if(receptor[a+4]==2)  
{  
    cont[2] = cont[2]+1;  
    tipo[2] = tipo[2] + receptor[a+14];  
}  
else if(receptor[a+4]==3)  
{  
    cont[3] = cont[3]+1;  
    tipo[3] = tipo[3] + receptor[a+14];  
}  
else if(receptor[a+4]==4)  
{  
    cont[4] = cont[4]+1;  
    tipo[4] = tipo[4] + receptor[a+14];  
}  
else if(receptor[a+4]==5)  
{  
    cont[5] = cont[5]+1;  
    tipo[5] = tipo[5] + receptor[a+14];  
}  
else if(receptor[a+4]==6)  
{  
    cont[6] = cont[6]+1;  
    tipo[6] = tipo[6] + receptor[a+14];  
}  
else if(receptor[a+4]==7)  
{  
    cont[7] = cont[7]+1;  
    tipo[7] = tipo[7] + receptor[a+14];  
}  
else if(receptor[a+4]==8)  
{  
    cont[8] = cont[8]+1;  
    tipo[8] = tipo[8] + receptor[a+14];  
}  
else if(receptor[a+4]==9)  
{  
    cont[9] = cont[9]+1;  
    tipo[9] = tipo[9] + receptor[a+14];  
}  
else if(receptor[a+4]==10)  
{  
    cont[10] = cont[10]+1;  
    tipo[10] = tipo[10] + receptor[a+14];  
}  
else if(receptor[a+4]==11)  
{  
    cont[11] = cont[11]+1;  
    tipo[11] = tipo[11] + receptor[a+14];  
}
```

```
//  
a=a+16;  
}  
Limpiar_vtemp();  
tot[1]=tot[1]+cont[1];  
vtemp[1]=tipo[1]/cont[1];  
vel[1]=chequeo (cont[1],vel[1],tipo[1]);  
tot[2]=tot[2]+cont[2];  
vtemp[2]=tipo[2]/cont[2];  
vel[2]=chequeo (cont[2],vel[2],tipo[2]);  
tot[3]=tot[3]+cont[3];  
vtemp[3]=tipo[3]/cont[3];  
vel[3]=chequeo (cont[3],vel[3], tipo[3]);  
tot[4]=tot[4]+cont[4];  
vtemp[4]=tipo[4]/cont[4];  
vel[4]=chequeo (cont[4],vel[4], tipo[4]);  
tot[5]=tot[5]+cont[5];  
vtemp[5]=tipo[5]/cont[5];  
vel[5]=chequeo (cont[5],vel[5], tipo[5]);  
tot[6]=tot[6]+cont[6];  
vtemp[6]=tipo[6]/cont[6];  
vel[6]=chequeo (cont[6],vel[6], tipo[6]);  
tot[7]=tot[7]+cont[7];  
vtemp[7]=tipo[7]/cont[7];  
vel[7]=chequeo (cont[7],vel[7], tipo[7]);  
tot[8]=tot[8]+cont[8];  
vtemp[8]=tipo[8]/cont[8];  
vel[8]=chequeo (cont[8],vel[8], tipo[8]);  
tot[9]=tot[9]+cont[9];  
vtemp[9]=tipo[9]/cont[9];  
vel[9]=chequeo (cont[9],vel[9], tipo[9]);  
tot[10]=tot[10]+cont[10];  
vtemp[10]=tipo[10]/cont[10];  
vel[10]=chequeo (cont[10],vel[10], tipo[10]);  
tot[11]=tot[11]+cont[11];  
vtemp[11]=tipo[11]/cont[11];  
vel[11]=chequeo (cont[11],vel[11], tipo[11]);  
if (mandar==1)  
{//velocidad promedio de todos los vehículos que se detectaron en los 5 min.  
vpromedio=((tipo[1]+tipo[2]+tipo[3]+tipo[4]+tipo[5]+tipo[6]+tipo[7]+tipo[8]+tipo[9]+tipo[10]+tipo[11])/cont[1]+cont[2]+cont[3]+cont[4]+cont[5]+cont[6]+cont[7]+cont[8]+cont[9]+cont[10]+cont[11]));  
}  
SFRPAGE = CONFIG_PAGE;  
EA=1;  
acum=0;  
Limpiar_tipo();  
Limpiar_cont();  
}  
void Vueltas_tot(void)  
{d=1;  
while(d < 12)  
{transmitir[30+d]=(tot[d]/256);  
d=d+1;  
}
```

```
}

void ARRANCAR_RELOJ(void) //inicializacion del oscilador del reloj
{A0='0';A1='0';A2='0';                                //7F8
D7='1';                                              //W bit=1
A0='1';                                              //7F9
D7='0';                                              //ST=0
A1='1';                                              //7FB
D7='1';                                              //KS=1
A0='0';A1='0';                                //7F8
D7='0';                                              //W=0
wait_ms(2000);                                //2 SEGUNDOS
D7='1';                                              //W=1
A0='1';A1='1';                                //7FB
D7='0';                                              //KS=0
A2='1';A1='1';A0='1';                            //7FF
D7='0';D6='0';D5='0';D4='0';D3='0';D2='1';D1='1';D0='0'; //ANO=6
A2='1';A1='1';A0='0';                            //7FE
D7='0';D6='0';D5='0';D4='0';D3='1';D2='0';D1='0';D0='1'; //MES=9
A2='1';A1='0';A0='1';                            //7FD
D7='0';D6='0';D5='0';D4='0';D3='1';D2='1';D1='1';D0='1'; //DIA=15
A2='1';A1='0';A0='0';                            //7FC
D7='0';D6='0';D5='0';D4='0';D3='0';D2='1';D1='1';D0='0'; //DIA=VIERNES
A2='0';A1='1';A0='1';                            //7FB
D7='0';D6='0';D5='0';D4='1';D3='0';D2='0';D1='1';D0='1'; //HORA=19
A2='0';A1='1';A0='0';                            //7FA
D7='0';D6='0';D5='1';D4='1';D3='0';D2='0';D1='1';D0='0'; //MINUTOS=50
A2='0';A1='0';A0='1';                            //7F9
D7='0';D6='0';D5='0';D4='0';D3='0';D2='0';D1='0';D0='0'; //SEGUNDOS=0
A2='0';A1='0';A0='0';                            //7F8
D7='0';                                              //W=0
}
}

void LEER_HORA(void)//lectura de la hora en el reloj externo
{A2='0';A1='0';A0='0';                                //7F8
D6='1';                                              //R=1
A2='1';A1='1';A0='1';                            //7FF
ano=D6*64+D5*32+D4*16+D3*8+D2*4+D1*2+D0*1;
A2='1';A1='1';A0='0';                            //7FE
mes=D4*16+D3*8+D2*4+D1*2+D0*1;
A2='1';A1='0';A0='1';                            //7FD
dia=D4*16+D3*8+D2*4+D1*2+D0*1;
A2='0';A1='1';A0='1';                            //7FB
hora=D4*16+D3*8+D2*4+D1*2+D0*1;
A2='0';A1='1';A0='0';                            //7FA
minutos=D5*32+D4*16+D3*8+D2*4+D1*2+D0*1;
A2='0';A1='0';A0='0';                            //7F8
D6='0';                                              //R=0
}
```

IEEE 802.11ac Radio Transmission Technology
IEEE 802.11ax Evolution-Data Only

AHS: Advanced Cruise-Assist Highway System

AP: Access Point

ASCII: American Standard Code for Information Interchange

AT: Attention

AVB: IEEE 802.1 Time Sensitive Networking

CDMA: Code Division Multiple Access

CDDA: Compact Disc Digital Audio

Apéndice C

Glosario de Acrónimos

CDP: Content Delivery Protocol

CPRI: Cell Public Radio Interface

CSMA: Carrier Sense Multiple Access

CSMA/CA: CSMA with Collision Avoidance

CSMA/CD: CSMA with Collision Detection

EAD: Estaciones Automatizadas de Adquisición de Datos de Tráfico Automotor

EEPROM: Electrically-Erasable Programmable Memory

EPROM: Erasable Programmable Memory

ETSI: European Telecommunications Standards Institute

FCC: Federal Communications Commission

FDDI: Fiber Distributed Data Interface

FW: Firewall

1xRTT: 1 time Radio Transmission Technology

1xEV-DO: 1 time Evolution-Data Only

1GHz intelligent

1.GPRS: General Packet Radio Service - A-

1.GSM: Global System for Mobile Telecommunications

AHS: Advanced Cruise-Assist Highway System

AP: Access Point

ASCII: American Standard Code for Information Interchange

AT: Attention

IDE: Integrated Development Environment

ISO/IEC 95

-C-

ISP: Internet Service Provider

CAN: Control Area Network

CDMA: Code Division Multiple Access

CIDI: Centro de Investigación y Desarrollo de Ingeniería

CPE: Costumer Premises Equipment

CPU: Central Process Unit

-E-

EAN: Electronic Article Network

EAADTA: Estaciones Automatizadas de Adquisición de Datos de Tráfico Automotor

EEPROM: Electrically-Erasable Programmable Memory

EPROM: Electrically Programmable Memory

ETRA: European Transport Research Activities

GPRS: General Packet Radio Service

MPPT: Multiple Path Hopping

-F-

MFB: Maximum Line Between Failure

FW: Firewall

-G-

GHz: GigaHertz

GPRS: General Packet Radio Service

GSM: Global System for Mobile Telecommunications

-I-

I²C: Inter-Integrated Circuit

IDE: Integrated Development Environment

IS-95: Interim Standard 95

ISP: Internet Service Provider

-S-

SDA: Software de Administración de Control -K-Tráfico-

SIM: Service Identification

Kbps: Kilobits por segundo

SRAM: Sane Random Access Memory

-L-

-T-

LAN: Local Area Network

TCP/IP: Transmission Control Protocol/ Internet Protocol

TTL: Transistor-Transistor Logic

-M-

-D-

Mbps: Megabits por segundo

MMS: Multimedia Messaging Service

MPH: Millas Por Hora

MTBF: Medium Time Between Failures

UDC-AP: Unidad de Control y Almacenamiento Temporal

UpPTS: "pulsos de Transmisión de Archivo"

UMTS: Universal Mobile Telecommunications System

Tienda de Componentes Serial Bus

-P-

Tienda de Componentes Serial Bus

PISACOTA: Plataforma Integrada para el Seguimiento, Análisis y Control del Tráfico Automotor

PPP: Point-to-Point Protocol

Tienda de Componentes Serial Bus

-R-

RAM: Random Access Memory

ROM: Read-Only Memory

Tienda de Componentes Serial Bus

SDU: Service Data Unit

-S-

Tienda de Componentes Serial Bus

SDCTU: Sistema Distribuido de Control de Tráfico Urbano

SDU: Service Data Unit

SMS: Short Message Service

SRAM: Static Random Access Memory

-T-

TCP/IP: Transmission Control Protocol/ Internet Protocol

TTL: Transistor-Transistor Logic

-U-

UART: Universal Asynchronous Receiver Transmitter

UCC: Unidad de Control Central

UDCAT: Unidad de Control y Almacenamiento Temporal

UDV: Unidad de Detección Vehicular

UMTS: Universal Mobile Telecommunication System

USB: Universal Serial Bus

UTI: Unidad de Transmisión Inalámbrica

-V-

VPN: Virtual Private Network

-W-

WAN: Wide Area Network

WAP: Wireless Application Protocol

WiFi: Wireless Fidelity

WiMAX: Worldwide interoperability for Microwave Access

Layout de las tarjetas de circuito impreso

Apéndice D

Layout de las tarjetas de circuito impreso

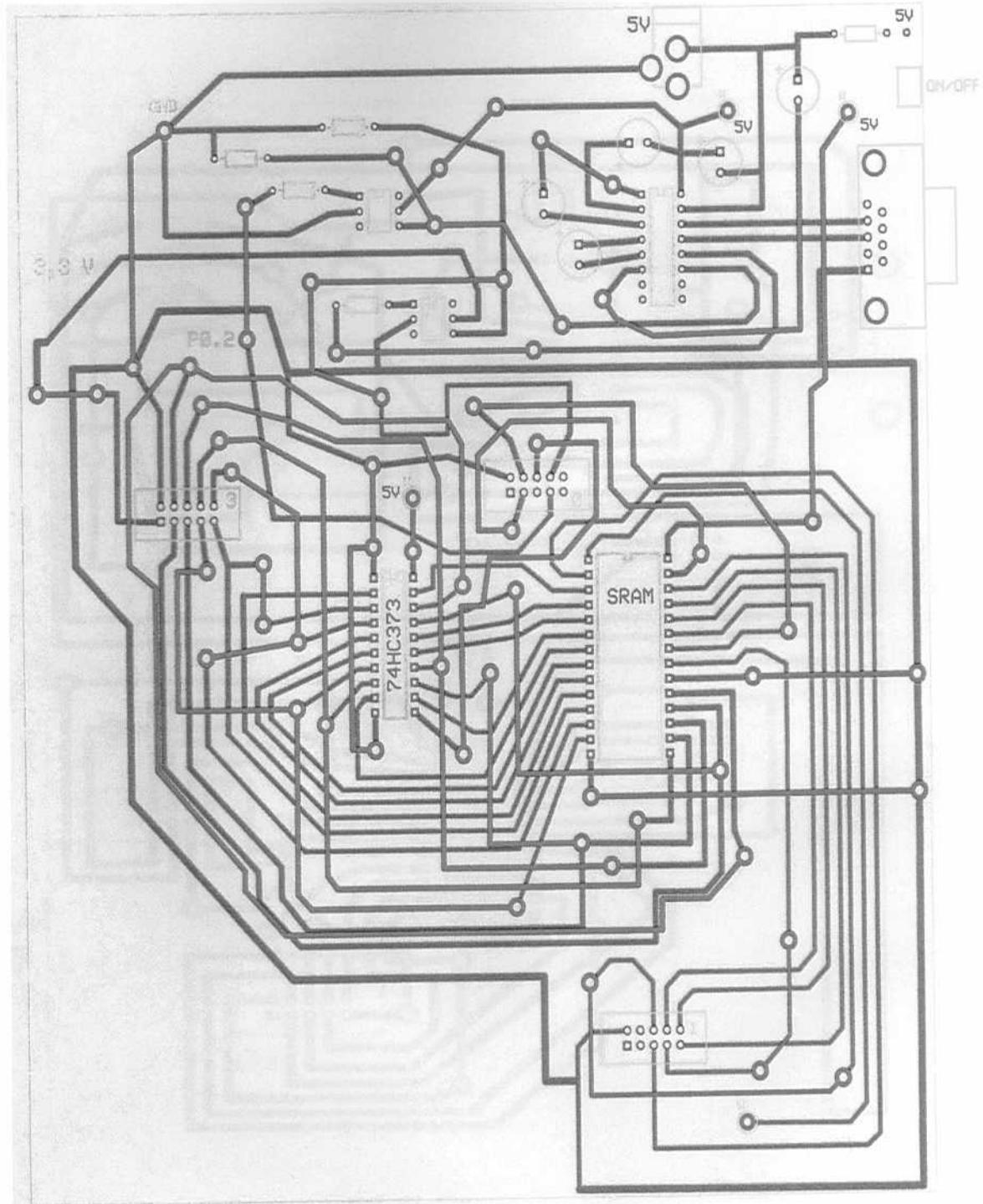


Figura D.1. Layout de la etapa de almacenamiento de la UDCAT

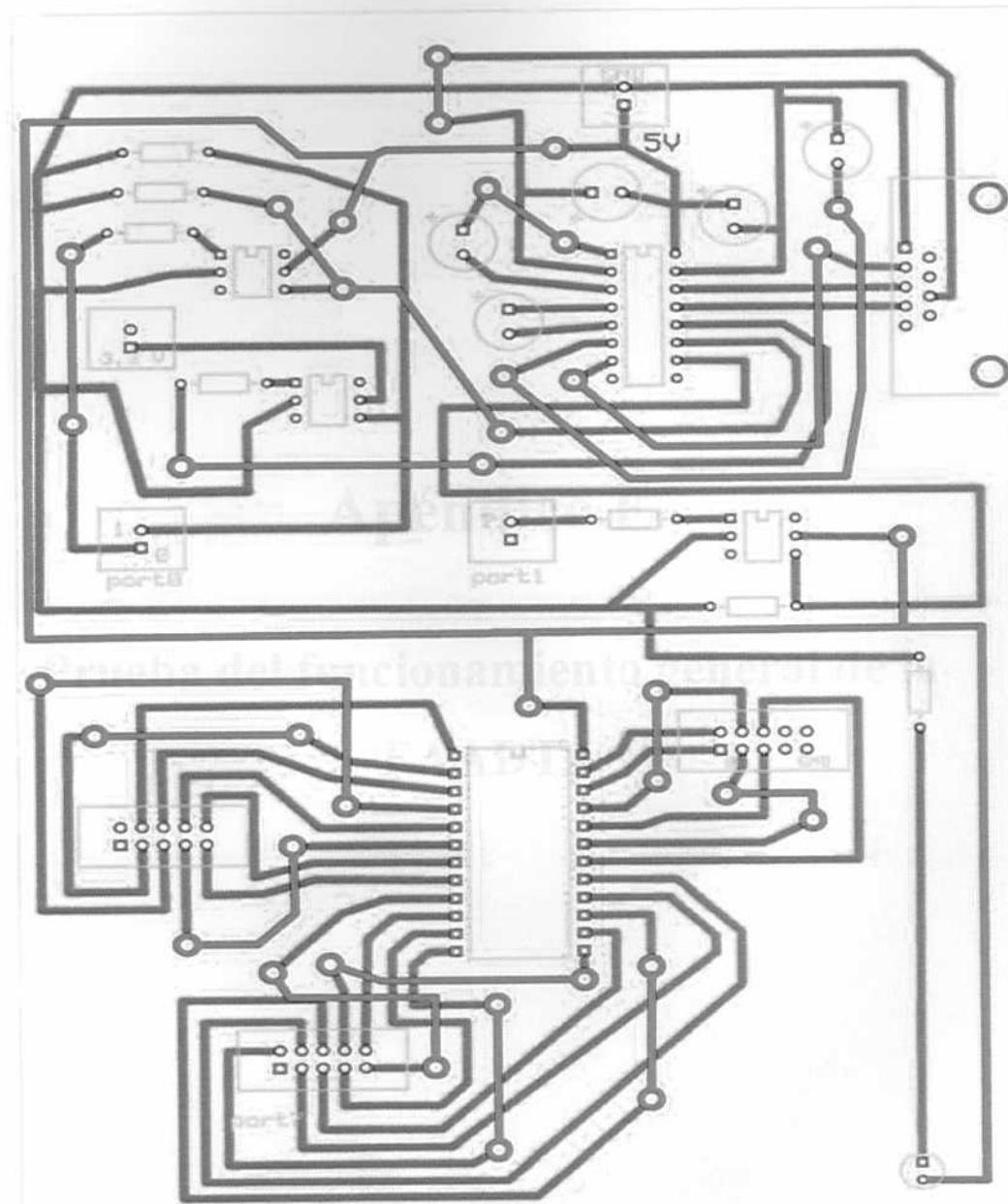


Figura D.2. Layout de la interfaz entre la UDCAT y la UTI junto con el reloj.

En la comunicación se muestran los parámetros que se han establecido en la configuración de la red. Se incluye la información de señal de respuesta al comando sobre el tipo de señal que se establecieron. La señal puede ser de alta o baja frecuencia.

Figura siguiente muestra la señal de respuesta de alta frecuencia:



Apéndice E

Prueba del funcionamiento general de la EAADTA



Figura 10.5. Tabla de datos de la EAADTA

A continuación se muestra una prueba del funcionamiento del sistema, en donde se exponen la información de salida que ofrece el sistema dependiendo los datos de entrada que se establecieron. La prueba consta de los siguientes pasos:

1. La Figura siguiente muestra la simulación realizada desde 7:58 am a las 8:03 am.

The screenshot shows a window titled "AUTOMATICO". Inside, there is a table with four columns: "# S", "# V", "Tipo", and "Velocidad (MPH)". The data is as follows:

# S	# V	Tipo	Velocidad (MPH)
1	45	1	30
2	5	2	22
3	87	3	64
4	12	4	55
5	32	5	49
6	11	6	43
7	17	7	44
8	16	8	37
9	6	9	30
10	2	10	31
11	2	11	24

Below the table are three buttons: "Agregar secuencia", "Enviar Data", and "Volver a Principal".

Figura E.1. Simulación desde 7:58 am a las 8:03 am

2. La Figura siguiente muestra la simulación realizada desde 8:03 am a 8:08 am.

The screenshot shows a window titled "AUTOMATICO". Inside, there is a table with four columns: "# S", "# V", "Tipo", and "Velocidad (MPH)". The data is as follows:

# S	# V	Tipo	Velocidad (MPH)
1	12	1	31
2	3	2	26
3	45	3	58
4	6	4	51
5	24	5	47
6	8	6	44
7	13	7	45
8	12	8	38
9	4	9	29
10	3	10	27
11	2	11	20

Below the table are three buttons: "Agregar secuencia", "Enviar Data", and "Volver a Principal".

Figura E.2. Simulación desde 8:03 am a las 8:08 am

3. La Figura siguiente muestra la simulación realizada desde 8:08 am a 8:13 am

The screenshot shows a software window titled "AUTOMATICO". Inside the window, there is a table with four columns: "# S", "# V", "Tipo", and "Velocidad [MPH]". The table contains 11 rows of data. Below the table is a button labeled "Volver a Principal". To the right of the table are two buttons: "Agregar secuencia" and "Enviar Data".

# S	# V	Tipo	Velocidad [MPH]
1	25	1	31
2	5	2	24
3	44	3	58
4	6	4	50
5	20	5	49
6	6	6	43
7	10	7	44
8	6	8	37
9	2	9	30
10	1	10	28
11	1	11	21

Figura E.3. Simulación desde 8:08 am a las 8:13 am

4. La Figura siguiente muestra la simulación realizada desde 8:13 am a 8:18 am

The screenshot shows a software window titled "AUTOMATICO". Inside the window, there is a table with four columns: "# S", "# V", "Tipo", and "Velocidad [MPH]". The table contains 11 rows of data. Below the table is a button labeled "Volver a Principal". To the right of the table are two buttons: "Agregar secuencia" and "Enviar Data".

# S	# V	Tipo	Velocidad [MPH]
1	35	1	37
2	7	2	25
3	56	3	57
4	8	4	52
5	24	5	53
6	9	6	45
7	13	7	47
8	14	8	35
9	6	9	32
10	2	10	29
11	1	11	23

Figura E.4. Simulación desde 8:13 am a las 8:18 am

5. Desde 8:18 am a 9:18 am se realizaron para cada período de 5 min una simulación como se muestra en la Figura siguiente.

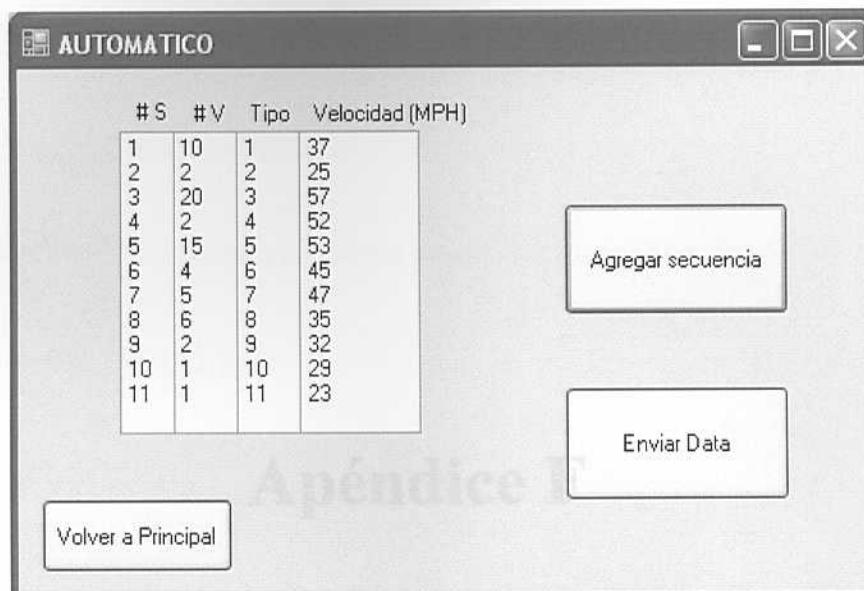


Figura E.4. Simulación de cada período de 5 min contenido en el lapso desde 8:18 am a 9:18 am

6. La Figura siguiente muestra el reporte visto a las 9:22am.

EAADTA: Parámetros de Tráfico											
ID	Estado	Fecha de Reporte (D/M/A)			Hora de Reporte (H/M)		Tipo 1		V. Tipo 1	Tipo 2	V. Tipo 2
1	1	9	9	2006	8	3	45	30	5	22	
1	1	9	9	2006	8	8	12	31	3	26	
1	1	9	9	2006	8	19	60	37	12	25	
1	3	9	9	2006	9	21	120	37	24	25	
Tipo 3	V. Tipo 3	Tipo 4	V. Tipo 4	Tipo 5	V. Tipo 5	Tipo 6	V. Tipo 6	Tipo 7	V. Tipo 7	Tipo 8	
87	64	12	55	32	49	11	43	17	44	16	
45	58	6	51	24	47	8	44	13	45	12	
100	57	14	52	44	53	15	45	23	47	20	
240	57	24	52	180	53	48	45	60	47	72	
V. Tipo 8	Tipo 9	V. Tipo 9	Tipo 10	V. Tipo 10	Tipo 11	V. Tipo 11	Vel Promedio	Nº Carros Total			
37	6	30	2	31	2	24	48,336171	235			
38	4	29	3	27	2	20	46,393939	132			
35	8	32	3	29	2	23	46,451429	299			
35	24	32	12	29	12	23	47,058888	826			

Figura E.5. Reporte expuesto en Excel visto a las 9:22am

AT <--> AT&T

AT

AT+CRM=130

AT

AT*PID=\$1763@0@camtv.net

OK

AT*ppw="*****" (Contraseña secreta proporcionada por Microfinc)

OK

AT+DIP="201.211.132.159"

AT

AT+INITR= "5050"

Apéndice F

Secuencia de comandos AT entre la UDCAT y el Módulo AnyData EMII-800

AT <--> AT&T

AT <--> AT&T (Cuando existe una conexión exitosa)

AT

AT <--> AT&T

AT <--> AT&T

Nota: El color negro representa los comandos enviados por la UDCAT.

No hay resaltado en la respuesta por parte del Módulo.

AT+CAD?

+CAD: 1

OK

AT+CRM=130

OK

AT*PID=5176360@cantv.net

OK

AT*PPW="*****" (Contraseña secreta proporcionada por Movilnet)

OK

AT+DIP="201.211.132.159"

OK

AT+DPORT="5050"

OK

ATDT123

Datasheet del Autosense II

Opción 1 (Cuando no logra conectarse a la red)

CALL

NO CARRIER

Opción 2 (Cuando existe una conexión exitosa)

CALL

PPP

CONNECT

Nota: El color negro representa los comandos enviados por la UDCAT y el color Naranja representa la respuesta por parte del Módulo.

AUTOSENSE II™
VEHICLE DETECTION
CLASSIFICATION SENSOR
User's Guide

Anexo A

Datasheet del Autosense II



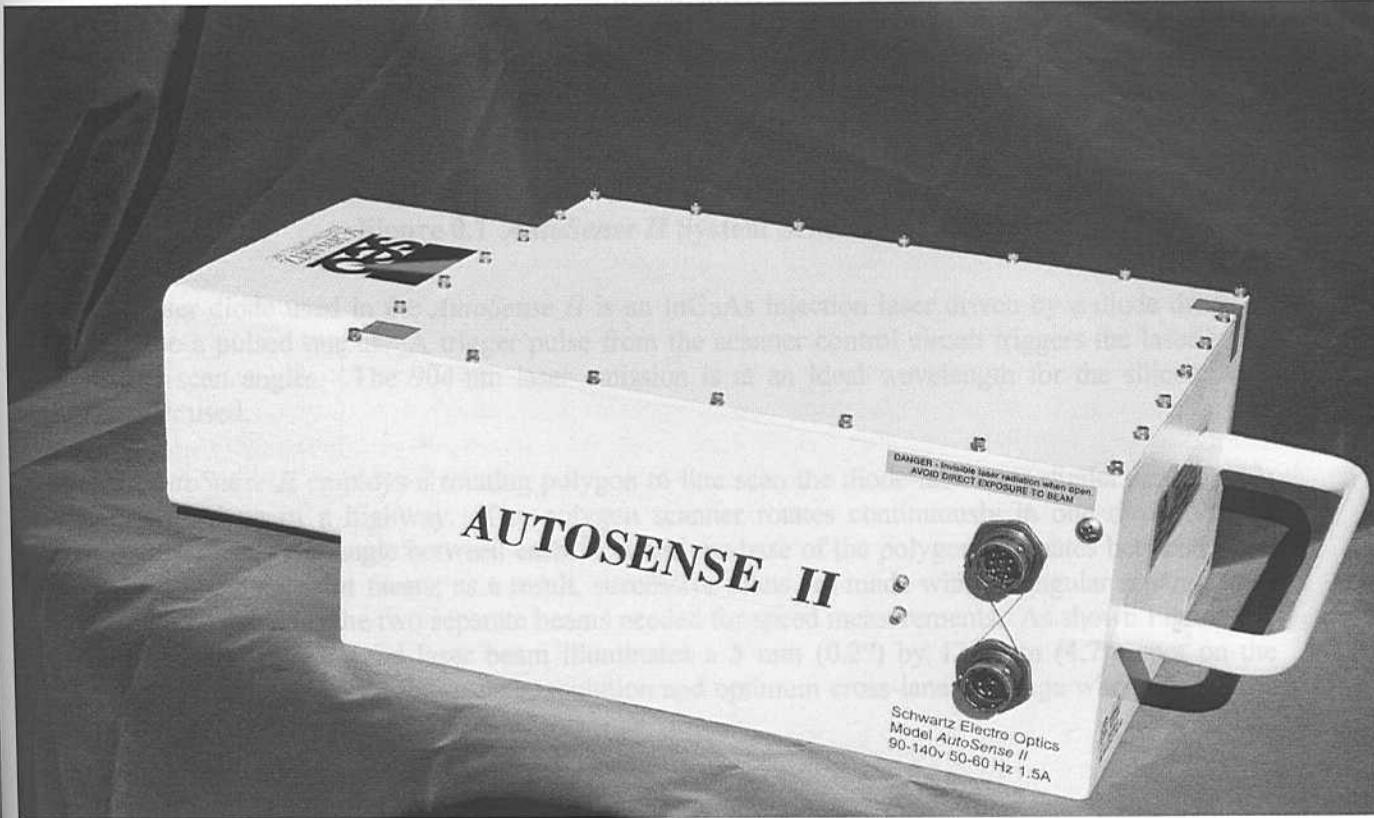
DESCRIPTION OF OPERATION

A schematic diagram of the AutoSense II system is shown in Figure 6. The AutoSense II laser rangefinder employs an InGaAs diode-laser transmitter and a silicon avalanche photodiode receiver in a side-by-side configuration. The transmitter optics of the diode laser assembly include a lens circuit and a collimating lens. The receiver assembly includes a lens, a beam splitter, a polarizer, an optical filter, detector/amplifier, and a microcontroller.

AUTOSENSE IITM

VEHICLE DETECTION CLASSIFICATION SENSOR

User's Guide



SECTION 0 DESCRIPTION OF OPERATION

A schematic diagram of the *AutoSense II* system is shown in Figure 0.1. The *AutoSense II*'s laser rangefinder employs an InGaAs diode-laser transmitter and a silicon avalanche photodiode (APD) receiver in a side-by-side configuration. The transmitter consists of the diode laser and its driver circuit and a collimating lens. The optical receiver is comprised of an objective lens, narrow-band optical filter, detector/amplifier, and threshold detector.

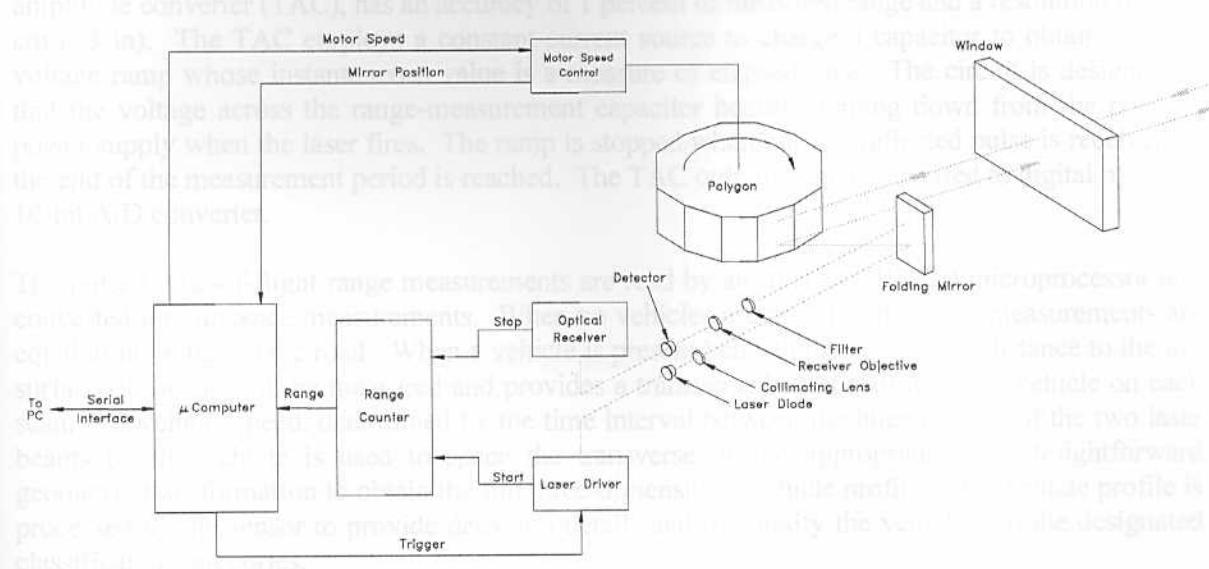


Figure 0.1 *AutoSense II* System Schematic Diagram

The laser diode used in the *AutoSense II* is an InGaAs injection laser driven by a diode driver to produce a pulsed output. A trigger pulse from the scanner control circuit triggers the laser at the proper scan angles. The 904-nm laser emission is at an ideal wavelength for the silicon APD receiver used.

The *AutoSense II* employs a rotating polygon to line scan the diode-laser rangefinder across a 12-foot-wide lane of a highway. The polygon scanner rotates continuously in one direction at a constant speed. The angle between each facet and the base of the polygon alternates between 87.5° and 92.5° for adjacent facets; as a result, successive scans are made with an angular separation of 10° , which provides the two separate beams needed for speed measurements. As shown Figure 0.4, the 0.5 mrad by 12 mrad laser beam illuminates a 5 mm (0.2") by 120 mm (4.7") spot on the pavement that provides good in-lane resolution and optimum cross-lane coverage when the laser is pulsed once per degree of scan angle.

DESCRIPTION OF OPERATION

The optical detection circuitry converts optical radiation reflected from the vehicle/road to first, an equivalent electrical analog of the input radiation and finally, a logic-level signal. The logic-level signals are processed within the range counter logic to yield analog range data, which is read by the microprocessor.

An analog range-measurement technique was chosen for the *AutoSense II* because of its better resolution, smaller size, simpler circuitry, lower power consumption, and lower cost when compared with digital techniques. The analog range measurement circuit, known as a time-to-amplitude converter (TAC), has an accuracy of 1 percent of measured range and a resolution of ± 7.6 cm (± 3 in.). The TAC employs a constant-current source to charge a capacitor to obtain a linear voltage ramp whose instantaneous value is a measure of elapsed time. The circuit is designed so that the voltage across the range-measurement capacitor begins ramping down from the positive power supply when the laser fires. The ramp is stopped when either a reflected pulse is received or the end of the measurement period is reached. The TAC output is then converted to digital by a fast 10-bit A/D converter.

The pulsed time-of-flight range measurements are read by an Intel 87C196KD microprocessor and converted into distance measurements. When no vehicles are present, the range measurements are equal to the range to the road. When a vehicle is present beneath the sensor, the distance to the top surface of the car will be measured and provides a transverse height profile of the vehicle on each scan. The vehicle speed, determined by the time interval between the interceptions of the two laser beams by the vehicle is used to space the transverse profile appropriately by straightforward geometric transformation to obtain the full three-dimensional vehicle profile. This vehicle profile is processed by the sensor to provide detection details and to classify the vehicle into the designated classification categories.

The *AutoSense II* employs a scanning laser rangefinder to measure three-dimensional vehicle profiles that can be used for very accurate vehicle classification. The narrow laser beam width permits the detection of closely spaced vehicles moving at high speed; even a two-inch-wide tow bar can be detected. The *AutoSense II* is ideal for applications involving electronic toll collection from vehicles at freeway speeds, where very high detection and classification accuracy is mandatory. The *AutoSense II* relies on an inherent laser characteristic -- narrow angular beam width -- to provide the high resolution required for accurate vehicle profiling. The *AutoSense II* beam-scan geometry is shown in Figure 0.2. The system scans two narrow laser beams, at a fixed angular separation, across the width of a lane at a rate of 720 scans per second. Pulsed time-of-flight range measurements provide accurate ± 7.6 cm (± 3 in.) transverse height profiles of a vehicle on each scan. The vehicle speed, determined from the time interval between the interceptions of the two laser beams by the vehicle, is used to space the transverse profiles appropriately to obtain the full three-dimensional vehicle profile. An algorithm, similar to those developed for military target recognition, is applied to the three-dimensional profile for vehicle-classification purposes.

DESCRIPTION OF OPERATION

dimensional profile generated for the coverage area is produced by the sensor to provide vehicle detection clearly and to classify the vehicle into the designated classification code.

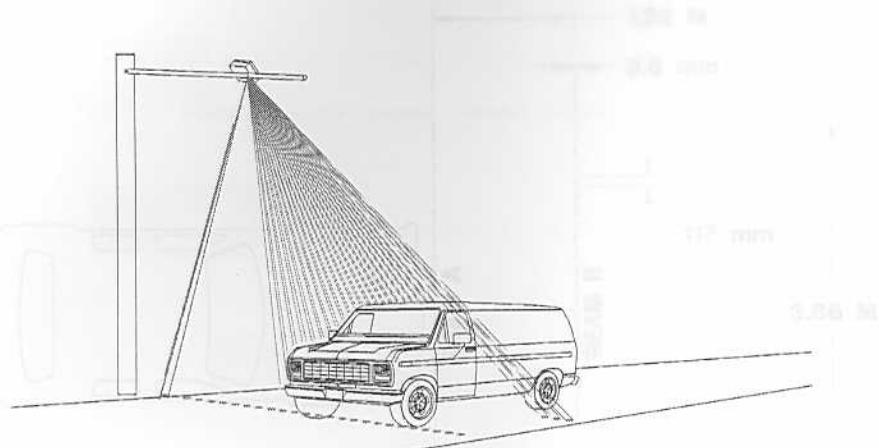


Figure 0.2: AutoSense II Beam Scan Geometry

An example of the *AutoSense II*'s three-dimensional profiling capability is provided by the range images shown in Figure 0.3. This range image of a van pulling a boat traveling at a speed of 45 mph was obtained by the *AutoSense II* operating with a scan rate of 360 scan/s. The pixel spacing resulting from the 1° scan resolution is more than adequate for vehicle identification purposes.

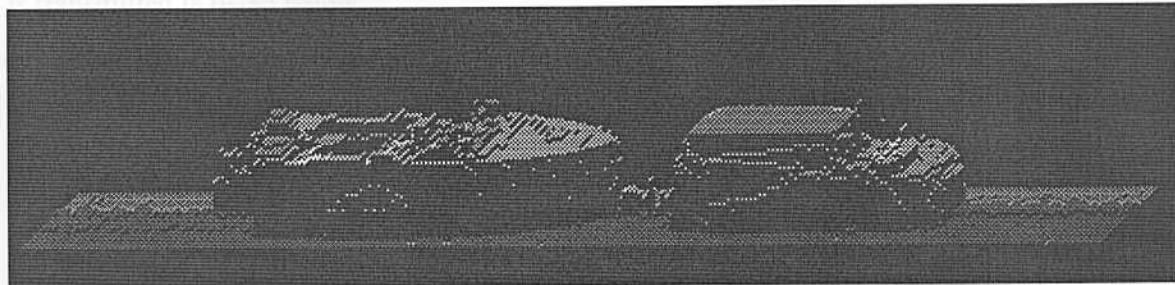


Figure 0.3: 3-D Range Image of Van Towing Boat

As described above, the *AutoSense II* employs a rotating polygon with alternating facet angles to achieve the fixed angular separation needed to line scan the diode-laser rangefinder across a lane of a highway. Alternating facet angles on the polygon allows the laser beam to trace the two lines across the road as the polygon rotates. Each range measurement for this laser beam illuminates a 5 mm (0.2") by 120 mm (4.7") spot on the pavement, as shown in Figure 0.4, that provides 3.66 meter (12 feet) total coverage when mounted 7 meters (23 feet) above the roadway. The three-

DESCRIPTION OF OPERATION

dimensional profile generated for this coverage area is processed by the sensor to provide vehicle detection details and to classify the vehicle into the designated classification categories.

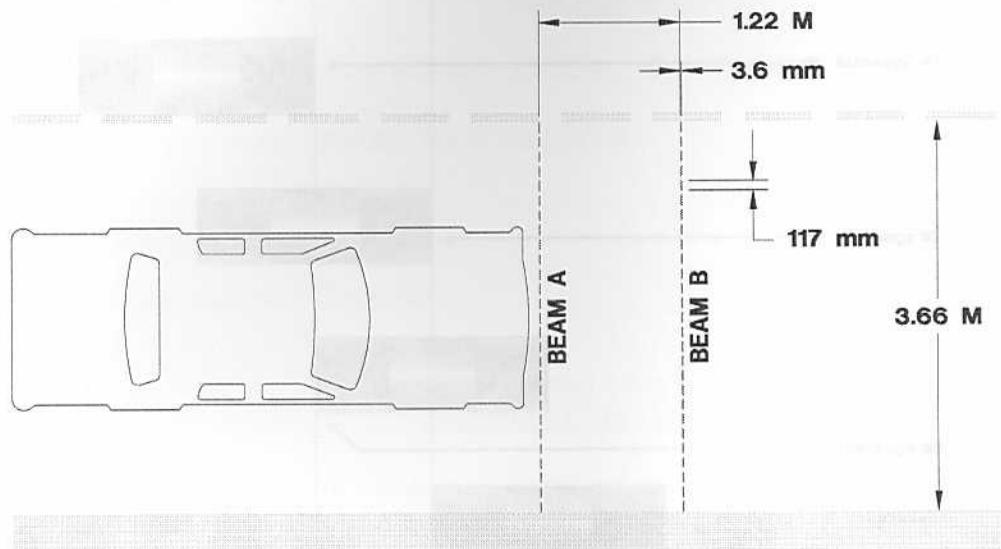


Figure 0.4: Laser Beam Footprints on Road Surface

The *AutoSense II* transmits five messages when operating in the detection mode for each vehicle detected within its field of view. In normal circumstances, each message and the order in which it is transmitted is listed below:

Figure 0.5: AutoSense Output Messages

- #1 First Beam Vehicle Detection Message
- #2 Second Beam Vehicle Detection Message
- #3 First Beam End of Vehicle Message
- #4 Second Beam End of Vehicle Message
- #5 Vehicle Classification Message

An illustration of the vehicle's position for each message is shown in Figure 0.5. The first four messages uniquely identify the vehicle and its position in the lane. The fifth message is the final message for the vehicle that includes vehicle classification, classification confidence percentage, height, length, width and speed.

The diagram illustrates the vehicle's position relative to the two laser beams. The vehicle is shown crossing the second beam. A serial message referred to as "Trigger Report" (message #5) is transmitted indicating an exit detection. The start of the Trigger Report message is transmitted within 6 ms from the time the vehicle clears the second beam.

DESCRIPTION OF OPERATION

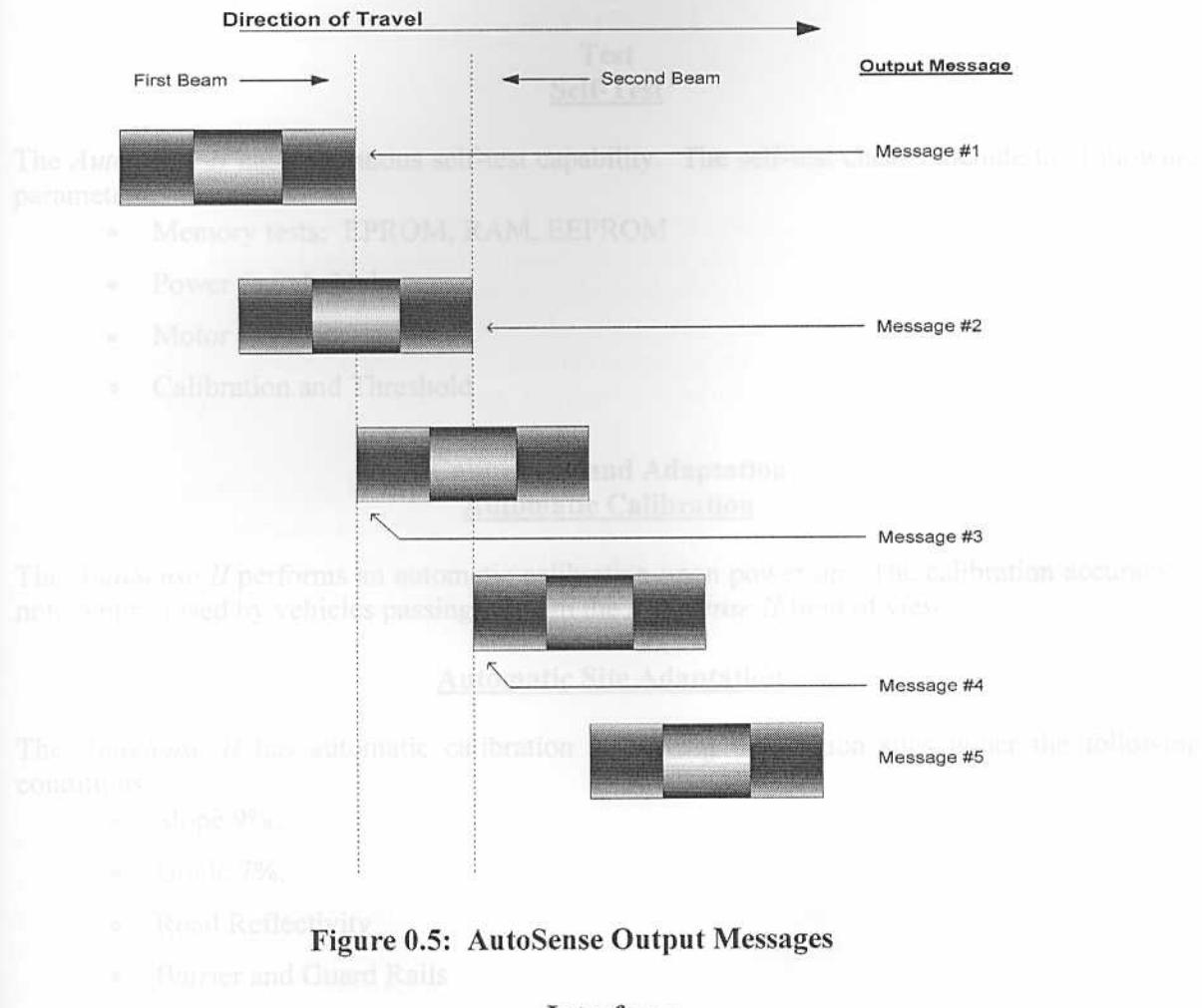


Figure 0.5: AutoSense Output Messages

Interfaces Serial Data Link

The *AutoSense II* provides an RS-422 (or RS-232) serial interface operating at a user selectable data rate of 19.2, 38.4 or 57.6 kilobits per second, 8 data bits, 1 start, 1 stop, no parity. The default data rate for normal operation is 57.6 Kilobits per second. The user selectable demonstration mode utilizes RS-422 only operating at a data rate of 1.25 Mbaud.

Exit Detection Trigger

The *AutoSense II* provides a real time signal when the vehicle's rear bumper has cleared the second laser beam. A serial message defined as a 'Trigger Report' (message 4 of 5) is transmitted indicating an exit detection. The start of the Trigger Report message begins transmission within 6 ms from the time the vehicle clears the second beam.

Test Self-Test

The *AutoSense II* has continuous self-test capability. The self-test checks include the following parameters.

- Memory tests: EPROM, RAM, EEPROM
- Power Supply Voltages
- Motor Control
- Calibration and Threshold

Calibration and Adaptation Automatic Calibration

The *AutoSense II* performs an automatic calibration upon power up. The calibration accuracy is not compromised by vehicles passing through the *AutoSense II* field of view.

Automatic Site Adaptation

The *AutoSense II* has automatic calibration at varying installation sites under the following conditions.

- Slope 9%.
- Grade 7%.
- Road Reflectivity
- Barrier and Guard Rails

1.2.1 Case.

The *AutoSense II* sensor and control system are housed in a durable case.

1.2.2 Power Input Connector (3-Pin).

Located on the side of the case is the 3-pin power input connector. The connector provides a connection to the power source. Two standard styles of connector are permitted. See paragraph 2.6.6.

1.2.3 Communication Connector (9-Pin).

Located above the power input connector is the 9-pin communication connector. This connector provides a connection to the communication center. Functional details of the connector are provided in section 2.7.1.

1.1.4 Red and Green Indicators**SECTION I****INTRODUCTION**

Located next to the connectors are two indicator lights. The red light indicates the AutoSense II™ is active and the green light indicates the AutoSense II™ is operating normally.

1.1 SCOPE.

The AutoSense II™ vehicle detection and classification sensor, as shown in figure 1.1, is a class I laser system that is ideally suited to providing toll and traffic management authorities with vehicle detection, presence, separation and classification information. A single sensor can be mounted directly over the traffic lane to be monitored (i.e., traditional toll plaza) or used in combination with several AutoSense II sensors mounted over several adjacent traffic lanes (i.e., open road applications). Please see Section 3 and Appendix E for detailed installation information.

The AutoSense II™ scans the roadway, taking range measurements across the width of the road at two locations beneath the sensor. These measurements are processed to generate messages that uniquely detect and classify each vehicle, and give its speed and position in the lane. The AutoSense II™ automatically initializes the vehicle detection process upon power-up, and its self-calibration process eliminates the need for any field adjustments.

This User's Guide covers safe and correct setup and operation of the AutoSense II™.

1.2 PHYSICAL DESCRIPTION.

The descriptions provided here are limited to support understanding of how to use and operate the AutoSense II™. For a detailed description, refer to Appendix A, Specification. There is no authorized corrective maintenance of the AutoSense II™ for user/operator performance. Access to the inside of the case is restricted to OSI-trained and authorized personnel. No description of the inside of the case is contained in this User's Guide. Figure 1-1 shows the AutoSense II™ in the fully operational configuration.

1.2.1 Case.

The AutoSense II™ laser and control system are housed in a durable case.

1.2.2 Power Input Connector (3-Pin).

Located on the side of the case is the 3-pin power input connector. The connector provides for connection of the power input cable. Functional details of connector are provided in paragraph 2.6.1.

1.2.3 Communication Connector (10-Pin).

Located above the power input connector is the 10-pin communication connector. The connector provides for connection of the communication cable. Functional details of connector are provided in paragraph 2.6.2.

1.2.4 Red and Green Indicators.

Located next to the connectors are a Red and a Green LED indicator. These indicators flash when the AutoSense II™ performs self-test at start up and are on during operation.

1.2.5 Laser Output Window.

Located at the front of the case, the laser output window protects the AutoSense II™ components from the environment, and allows the laser to scan the traffic lane.

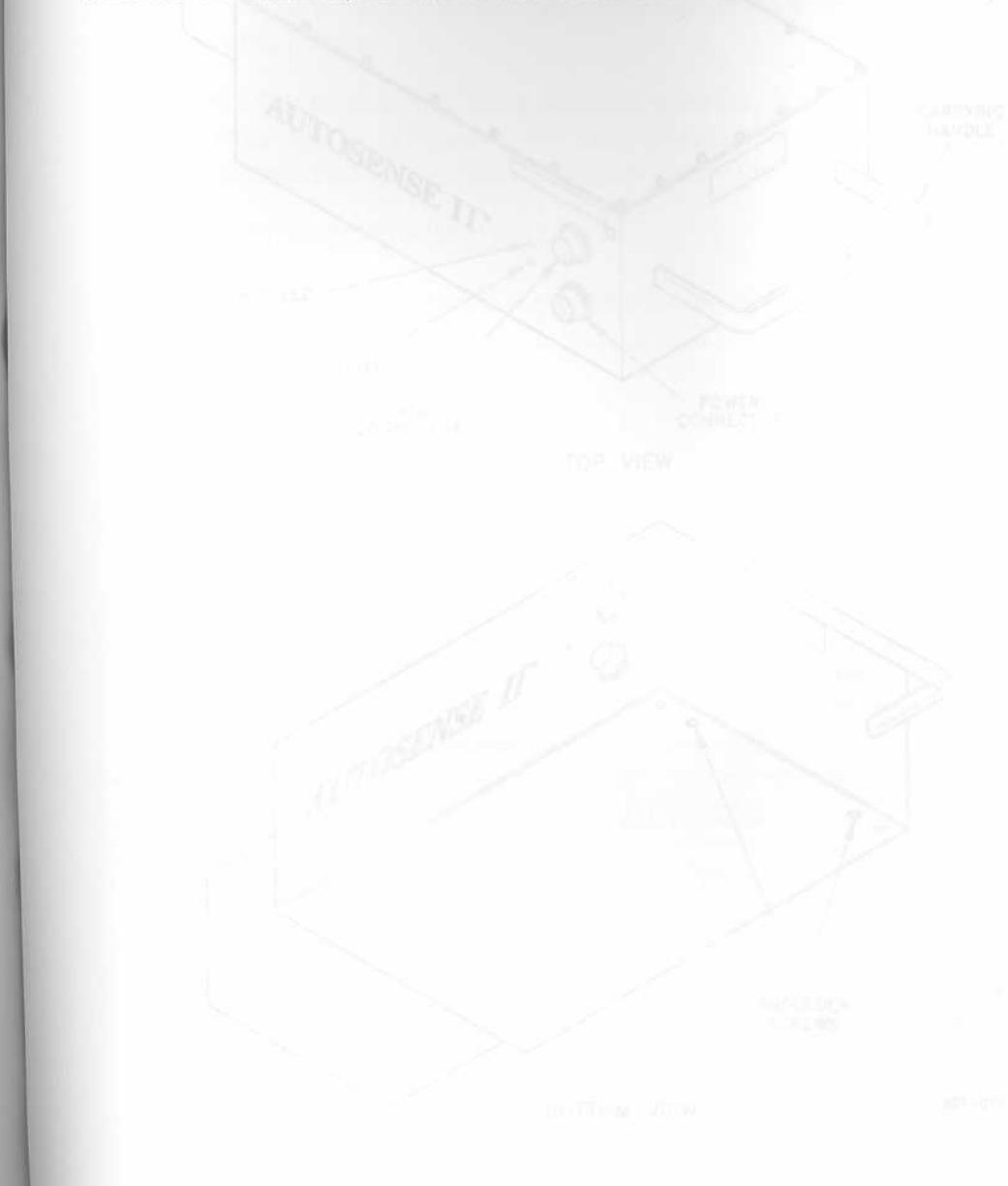


Figure 1. AutoSense II™.

INTRODUCTION

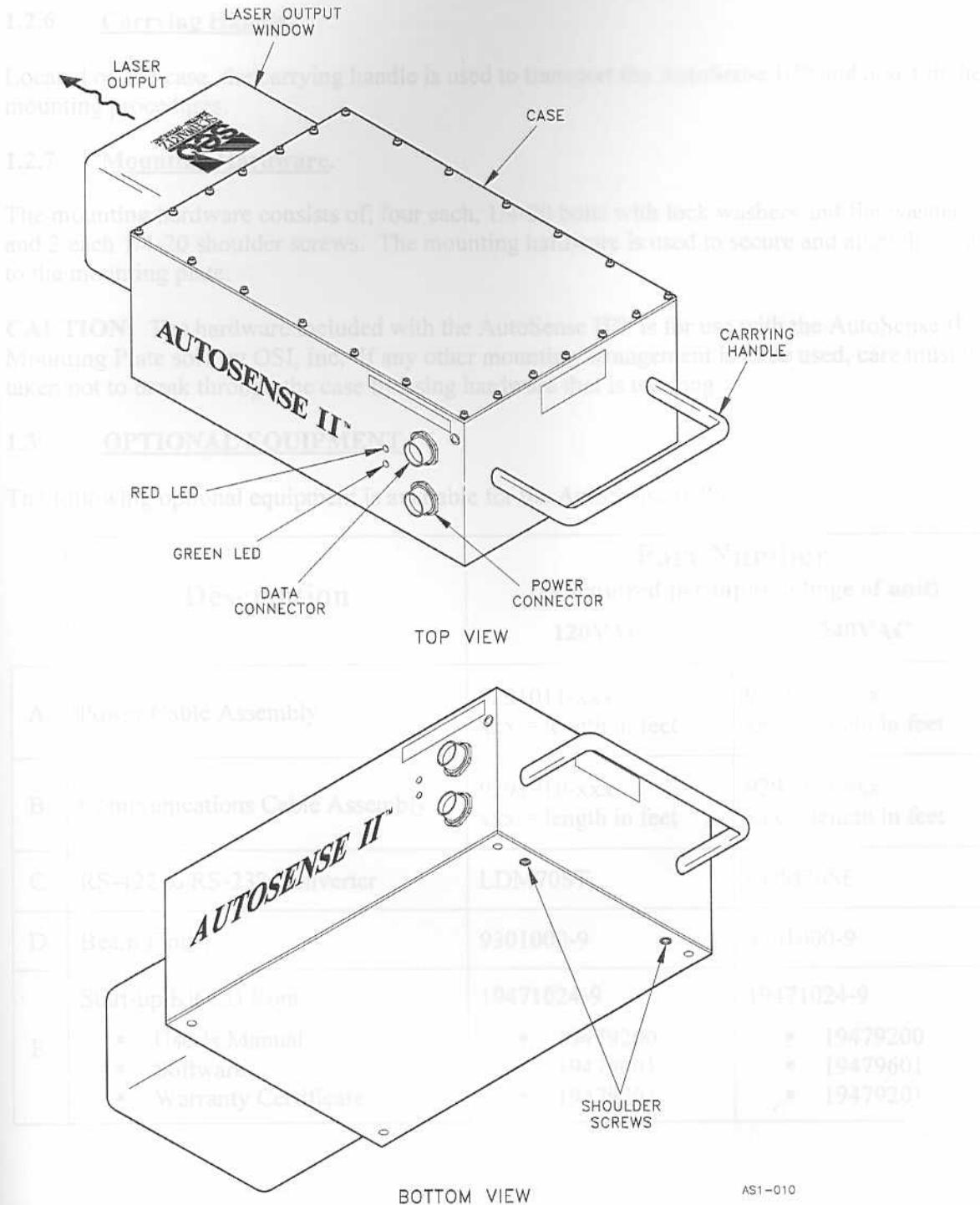


Figure 1-1. AutoSense II™.

1.2.6 Carrying Handle.

Located on the case, the carrying handle is used to transport the AutoSense II™ and assist in the mounting procedures.

1.2.7 Mounting Hardware.

The mounting hardware consists of; four each, 1/4-20 bolts with lock washers and flat washers; and 2 each 1/4-20 shoulder screws. The mounting hardware is used to secure and align the case to the mounting plate.

CAUTION: The hardware included with the AutoSense II™ is for use with the AutoSense II™ Mounting Plate sold by OSI, Inc. If any other mounting arrangement is to be used, care must be taken not to break through the case by using hardware that is too long.

1.3 OPTIONAL EQUIPMENT.

The following optional equipment is available for the AutoSense II™:

	Description	Part Number (as required per input voltage of unit)	
		120VAC	240VAC
A	Power Cable Assembly	9291011-xxx xxx = length in feet	9291111-xxx xxx = length in feet
B	Communications Cable Assembly	9291010-xxx xxx = length in feet	9291010-xxx xxx = length in feet
C	RS-422 to RS-232 Converter	LDM70ST	LDM70SE
D	Beam Finder	9301000-9	9301000-9
E	Start-up Kit CD Rom <ul style="list-style-type: none">▪ User's Manual▪ Software▪ Warranty Certificate	19471024-9 <ul style="list-style-type: none">▪ 19479200▪ 19479601▪ 19479201	19471024-9 <ul style="list-style-type: none">▪ 19479200▪ 19479601▪ 19479201

F	Mounting Kit ▪ Mounting plate ▪ Mounting hardware and tools ▪ Connector, Data ▪ Connector, Power	19471022-9 ▪ 19476022-1 ▪ 19471023-9 ▪ PW06P-12-10S ▪ PW06P-12-3S	19471022-19 ▪ 19476022-1 ▪ 19471023-9 ▪ PW06P-12-10S ▪ PW06P-12-3SY
G	Supplemental Communications Cable Surge Suppressor	81000143-9	81000143-9
H	RS422 DMA Board	86000426-1	86000426-1

The shipping container is provided by foamed inserts. The items listed below are located in the shipping container. Remove the components from the shipping container and inspect for damage.

- a. Place the shipping container down on a flat surface.
- b. Open the lid.

c. Verify the following components are in the shipping container and are not damaged.

1. AutoSense (IP) Unit

2. AutoSense Surge Suppressor

3. Four (4) #6-32 x 1/2" Lock Washers

4. Two (2) #6-32 x 1/2" Flat Washers

5. One (1) AutoSense (IP) Unit

6. Optional Items, as shown on the packing list

Packing List.

- a. Remove the items from the shipping container, place on a flat surface (table or counter), and inspect for damage.

2.3 **Power Requirements**

The AutoSense (IP) Unit is configured for one of two power requirements as follows:

a. 90-140 V, 50-60 Hz, 1.5 A

b. 100-264 V, 50-60 Hz, 1.5 A

3.1 **POWER**

Check the AutoSense (IP) power requirements listed on the case, prior to applying power.

F	Mounting Kit ▪ Mounting plate ▪ Mounting hardware and tools ▪ Connector, Data ▪ Connector, Power	19471022-9 ▪ 19476022-1 ▪ 19471023-9 ▪ PW06P-12-10S ▪ PW06P-12-3S	19471022-19 ▪ 19476022-1 ▪ 19471023-9 ▪ PW06P-12-10S ▪ PW06P-12-3SY
G	Supplemental Communications Cable Surge Suppressor	81000143-9	81000143-9
H	RS422 DMA Board	86000426-1	86000426-1

Component is provided by foamed inserts. The items used with the AutoSense II™ are listed below in their foam inserts. Unpack the AutoSense II™, as follows:

- a. Place the shipping cradles down on a flat surface.
- b. Open the lid.

- c. Make the following components are in the shipping cradles:

▪ 9.5 in. LCD Unit

▪ 40 Scan Under Survey

▪ Four 1/4-20 inch, lock Washers

▪ Two AutoSense II™ Laser Sensors

▪ Optional Items, as shown in the packing list

▪ Packing List.

- d. Remove the items from the shipping container, place on a flat work surface (table or counter) and inspect for damage.

2.3 Power Requirements

The AutoSense II™ is intended for one of two power requirements:

a. 90-130 V, 50-60 Hz, 1.5 A

b. 100-240 V, 50-60 Hz, 1.0 A

2.4 Power

Check the AutoSense II™ power requirements label on the case, prior to applying power.

2.4 **INTERFACE SIGNALS** SECTION 2

GETTING STARTED

The AutoSense II™ default serial port configuration is configured for RS-232C with 9600 baud.

For further information, signals for each of the interfaces are presented in Table 2-1.

2.1 **SCOPE.**

The following procedures provide instructions for unpacking, inspection, and general interface and power requirements of the AutoSense II™.

2.2 **UNPACKING AND INSPECTION OF PARTS.**

The AutoSense II™ is shipped in a container. Inside the container, extra protection of the equipment is provided by foamed inserts. The items used with the AutoSense II™ are imbedded in the foam inserts. Unpack the AutoSense II™, as follows:

- a. Place the shipping container down on a flat surface.
- b. Open the lid.
- c. Ensure the following components are in the shipping container; check Packing List against components:
 1. AutoSense II™, Unit
 2. Two Shoulder Screws
 3. Four 1/4-20 Bolts, four Lock Washers, four Flat Washers
 4. AutoSense II™ User's Guide
 5. Optional Items, as shown on the Packing List
 6. Packing List.
- d. Remove the items from the shipping container; place on a flat work surface (table or bench), and inspect for damage.

2.3 **POWER REQUIREMENTS.**

The AutoSense II™ is configured for one of two power requirements, as follows:

- a. 90-140 V, 50-60 Hz, 1.5 A
- b. 200-264 V, 50-60 Hz, 1.0 A.

CAUTION

Check the AutoSense II™ power requirements label on the case, prior to applying power.

2.4 INTERFACE SIGNALS.

The AutoSense II™ default serial interface is RS-422, but may be configured for RS-232. Both are full duplex. Signals for each of the interfaces are presented in Table 2-1.

Table 2-1. AutoSense II™ Interface Signals.

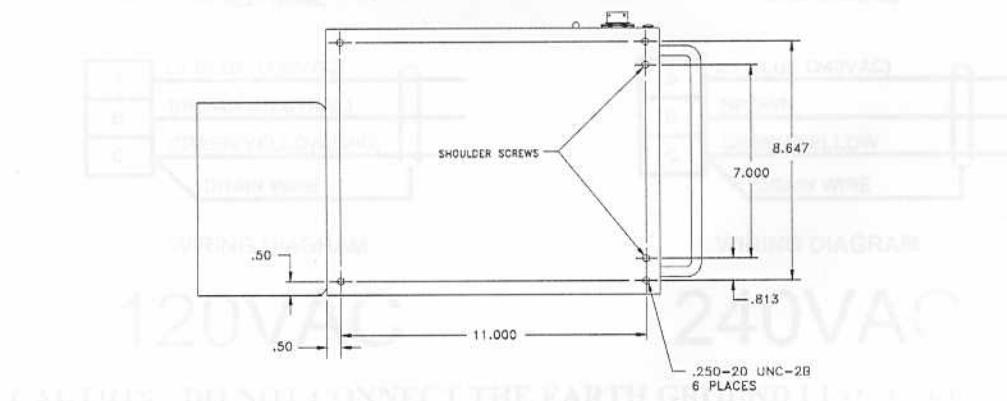
STANDARD CONFIGURATION	Description	Specification
	RS-422 *	
	Function	Serial data communication
	Mode of Operation	Differential
	Configuration	No Parity, 8 data bits, 1 stop bit
	Baud Rate	19.2, 38.4, 57.6 Kbaud
	Maximum Cable Length	Dependent upon data rate, \leq 3300 feet @ 57.6 Kbaud
	RS-232 (EIA 232D) *	
	Function	Serial data communication
	Mode of Operation	Single ended
	Configuration	No Parity, 8 data bits, 1 stop bit
	Baud Rate	19.2, 38.4, 57.6 Kbaud
	Maximum Cable Length	Dependent upon cable capacitance, 2500 pF Max, 165 feet typical

* Note: These are factory settings only

2.5 DIMENSIONS.

The dimensions of the AutoSense II™ are listed below. Figure 2-1 shows the mounting holes on the base of the case.

- a. Height: 5.8 inches (14.6 cm)
- b. Width: 9.6 inches (24.2 cm)
- c. Depth: 19.1 inches (48.5 cm)
- d. Weight 25 pounds (11 kg).

**Figure 2-1. AutoSense II™ Housing Dimensions (In Inches).**

2.6 CABLE REQUIREMENTS.**2.6.1 Power Cable.**

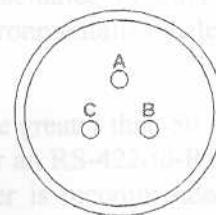
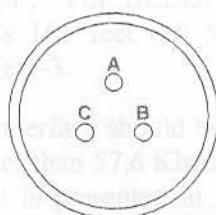
Input Power Connector pin functions are described in Table 2-2. The three-conductor power cable should be shielded and have a minimum wire size of 18 AWG. Alpha 5163/1C is an acceptable, recommended cable. OSI provides an environmentally sealed cable as an accessory. See page 1-4.

Table 2-2. Input Power Connector Pin Functions.

Pin	Function
A	AC
B	AC
C	Earth Ground

For proper installation, it is necessary to connect all three conductors to the specified power and ground connections. A fuse or circuit breaker should be used between the AutoSense II™ unit and the source of power.

If the cable shield is to be grounded it is recommended that this be done only on the source end of the cable.

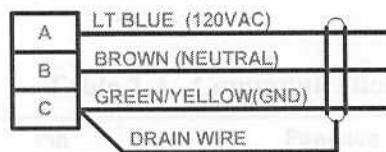


CONNECTOR P2 PIN-OUTS
ENLARGED VIEW SHOWN FROM
WIRING SIDE OF CONNECTOR

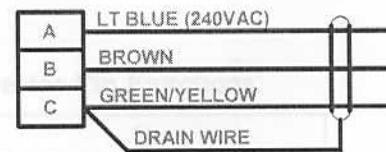
CONNECTOR P2 PIN-OUTS
ENLARGED VIEW SHOWN FROM
WIRING SIDE OF CONNECTOR

SCALE: NONE

SCALE: NONE



WIRING DIAGRAM



WIRING DIAGRAM

120VAC

240VAC

CAUTION DO NOT CONNECT THE EARTH GROUND LINE THROUGH A FUSE OR CIRCUIT BREAKER.

Connect the power cable plug to the AutoSense II™ power input connector, as shown in Figure 2-2.



Figure 2-2. Power Cable Connection.

A51-003

2.6.2 Communication Cable.

Communication Data Connector pin functions for the AutoSense II™ are described in Table 2-3. The cable used should be a shielded, low-capacitance, polyethylene type, such as BELDEN 9807. For RS-422 operation, maximum cable length for reliable operation is determined by data rate. For RS-232 total cable capacitance will determine the maximum cable length, and should not exceed 2500 pF. For BELDEN 9807, which has a capacitance of 50.4 pF/meter, the maximum length is 165 feet (50 M). OSI provides an environmentally sealed cable as an accessory. See page 1-3.

The RS-422 serial interface should be used when cable lengths are greater than 50 meters or when baud-rates are greater than 57.6 Kbaud. A connection diagram for an RS-422-to-RS-232 converter (Dataforth LDM70) is presented in Appendix B. This converter is recommended for use with computers that require an RS-232 interface.

AutoSense II also has a high-speed (1.25 Mbps) RS-422 interface capability. The high-speed interface is used in applications requiring transmission of the sensor's raw range and intensity data. A connection diagram for a high-speed RS-422 interface is presented in Appendix B.

Table 2-3. Communication Data Connector Pin Functions.

Pin	Function	Note
A	Ground	Twisted Pair
K	Camera Trigger	
B	RS-422 TX +	Twisted Pair
C	RS-422 TX -	
F	RS-422 RX +	Twisted Pair
G	RS-422 RX -	
H	RS-232 TX	Twisted

J	RS-232 RX	Pair
D	Cable Shield	
E	Unused	

3.1 FUNCTIONAL DESCRIPTION

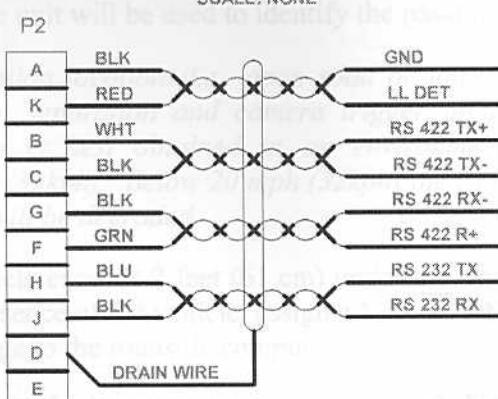
The AutoSense II™ provides timing information and classification information to the vehicle's field-of-view. The AutoSense II™ will be mounted to be installed in a fixed location, such as a highway, in an open road environment or mounted on a pole or other structure in a well place pointing down at the center of the vehicle's field-of-view. The AutoSense II™ will communicate with a roadside computer via a standard data connector, using either RS-232 or RS-422. It also provides a camera trigger signal through the same connector to the vehicle passing through its field-of-view. The AutoSense II™ will output five types of data: vehicle types and a camera trigger, as shown in Figure 2-2 and Table 2-1. The vehicle ID number assigned by the manufacturer will be used to identify the passing vehicle. The vehicle ID number assigned by the manufacturer will be used to identify the passing vehicle.

CONNECTOR P2 PIN-OUTS

ENLARGED VIEW SHOWN FROM FIGURE 2-2

WIRING SIDE OF CONNECTOR

SCALE: NONE

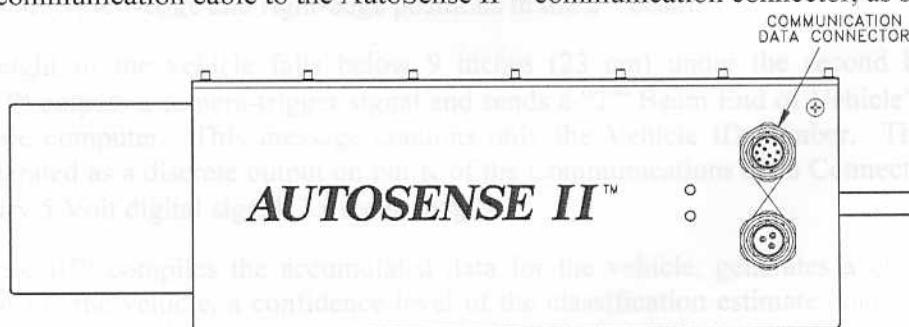


WIRING DIAGRAM

COMM INPUT

When the vehicle's field-of-view falls below 9 inches (23 cm) under the vehicle's beam, the vehicle's left edge will trigger a "near" message to the roadside computer.

Connect the communication cable to the AutoSense II™ communication connector, as shown in Figure 2-3.



The AutoSense II™ samples the accumulated data for the vehicle's field-of-view, and sends raw data to the roadside computer. The message contains the vehicle's field-of-view, a confidence level of the classification estimate, and the vehicle's left edge signal. The roadside computer processes the raw data and sends regular data to the vehicle.

Figure 2-3. Communication Cable Connection.

AS1-004

SECTION 3

OPERATION OF THE AUTOSENSE II™

3.1 FUNCTIONAL DESCRIPTION.

The AutoSense II™ provides timing, position, speed, and classification of vehicles passing through its field-of-view. The AutoSense II™ is designed to be installed on a gantry or pole above a highway, in an open road environment, or installed on a pole or other support structure in a toll plaza pointing down at the center of a traffic lane. The AutoSense II™ is designed to communicate with a roadside computer through its serial data connector, using either RS-232 or RS-422. It also provides a camera trigger signal as a discrete through the same connector. For each vehicle passing through its field-of-view, the AutoSense II™ will output five serial data messages and a camera trigger, as shown and described in Figure 3-1 and Table 3-1. A Vehicle ID number assigned by the unit will be used to identify the passing vehicle for all five messages.

Regardless of the installation location (i.e., open road or toll plaza) the AutoSense II™ will provide vehicle detection, separation and camera trigger information. However, accurate classification information is best obtained in an environment where vehicle speeds are maintained above 20 mph (32kph). Below 20 mph (32kph) the confidence factor associated with the vehicle classification will be degraded.

When the height of a vehicle exceeds 2 feet (61 cm) under the first beam of the AutoSense II™, the unit will detect the presence of the vehicle, assign a Vehicle ID number, and send a “1st Beam Vehicle Detection” message to the roadside computer.

When the height of the vehicle exceeds 2 feet (61 cm) beneath the second beam, the AutoSense II™ will send a “2nd Beam Vehicle Detection” message to the roadside computer. This message contains the Vehicle ID number, the vehicle speed and the vehicle’s left-edge and right-edge positions in the 1st beam.

When the height of the vehicle falls below 9 inches (23 cm) under the first beam, the unit will send a “1st Beam End of Vehicle” message to the roadside computer. This message contains the Vehicle ID number, left-edge and right-edge positions in the 2nd beam.

When the height of the vehicle falls below 9 inches (23 cm) under the second beam, the AutoSense II™ outputs a camera-trigger signal and sends a “2nd Beam End of Vehicle” message to the roadside computer. This message contains only the Vehicle ID number. The camera trigger is generated as a discrete output on pin K of the Communications Data Connector. It is a complementary 5 Volt digital signal, 1 μ sec in length..

The AutoSense II™ compiles the accumulated data for the vehicle, generates a classification code that best fits the vehicle, a confidence level of the classification estimate, and sends these data along with the Vehicle ID number, vehicle length, width, height and speed to the roadside computer. This is the “Classification Message” and is the last message the AutoSense II™ will send regarding the vehicle.

MESSAGE PROTOCOL

In addition, the AutoSense II™ runs self-tests on a periodic basis and will report any detected failures by sending a Self-Test Message to the roadside computer.

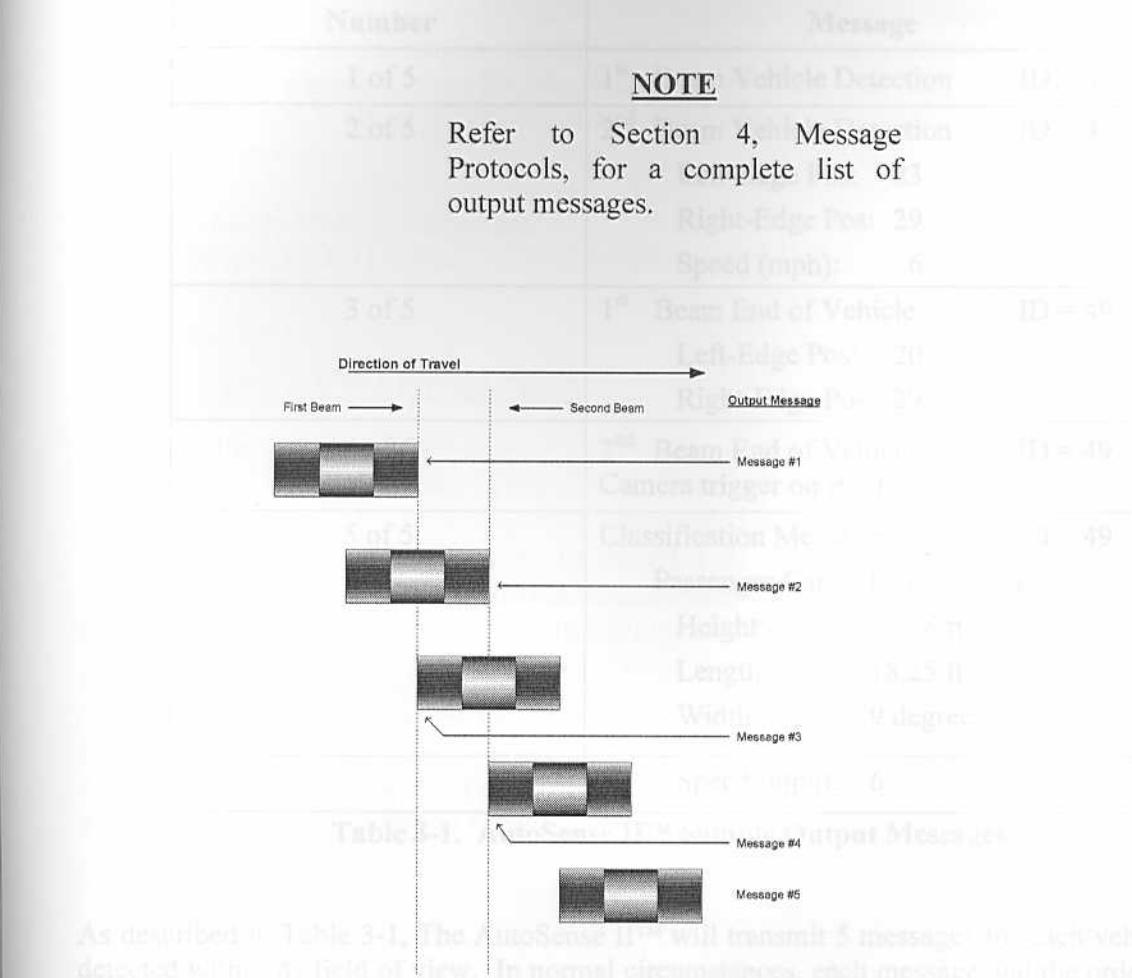


Figure 3-1. AutoSense II™ Output Messages Versus Vehicle Location.

- 1. First Beam End of Vehicle Message
- 2. Second Beam End of Vehicle Message
- 3. Vehicle Classification Message

Each message will include a corresponding vehicle ID. For a vehicle to be validated all three will must be received when the same vehicle ID. If any message is not received all of the messages do not have the same vehicle ID, then the vehicle is not valid and be counted.

Normal 5 message sequence:

[Msg#1] [ID#1] [Msg#2] [ID#1] [Msg#3] [ID#1] [Msg#4] [ID#1] [Msg#5]

Number	Message	
1 of 5	1 st Beam Vehicle Detection	ID = 49
2 of 5	2 nd Beam Vehicle Detection Left-Edge Pos: 23 Right-Edge Pos: 29 Speed (mph): 6	ID = 49
3 of 5	1 st Beam End of Vehicle Left-Edge Pos: 20 Right-Edge Pos: 29	ID = 49
4 of 5	2 nd Beam End of Vehicle Camera trigger on Pin K	ID = 49
5 of 5	Classification Message Passenger Car w/Trailer: 70% Height: 5.25 ft Length: 18.25 ft Width: 9 degrees Speed (mph): 6	ID = 49

Table 3-1. AutoSense II™ Sample Output Messages.

As described in Table 3-1, The AutoSense II™ will transmit 5 messages for each vehicle that is detected within its field of view. In normal circumstances, each message and the order in which it is transmitted is listed below.

- #1 First Beam Vehicle Detection Message
- #2 Second Beam Vehicle Detection Message
- #3 First Beam End of Vehicle Message
- #4 Second Beam End of Vehicle Message
- #5 Vehicle Classification Message

Each message will include a corresponding vehicle ID. For a vehicle to be validated, messages #1 through #5 must be received when the same vehicle ID. If any message is not received, or if all of the messages do not have the same vehicle ID, then the vehicle is not valid and should not be counted.

Normal Vehicle Message Sequence:

[Msg#1, ID=1] [Msg#2, ID=1] [Msg#3, ID=1] [Msg#4, ID=1] [Msg#5, ID=1]

However, if the AutoSense II™ unit is installed incorrectly and the 10 degree laser beam is not pointing into the oncoming flow of traffic, the AutoSense II™ will transmit messages out of the normal sequence. Likewise, if a vehicle were to back-up through the detection zone (i.e., travel in the opposite direction of traffic), the AutoSense II™ will also transmit the messages out of sequence. This will allow the off-board computer system to identify and filter this detection as an invalid detection.

For normal operation, the AutoSense II™ is configured for 57.6 Kbaud, no parity, 8 data bits, Reverse Vehicle Message Sequence: every headstart is used. The AutoSense II™ will send the sequence [Msg#2, ID=1] [Msg#1, ID=2] [Msg#4, ID=2] [Msg#5, ID=2] [Msg#3, ID=1] header followed by the message command, and ending with a 1 byte checksum, as shown in the following table. The sections that follow describe the message blocks for each of the AutoSense II™ messages. The following table is transferred low by high.

3.2 TYPICAL INSTALLATION.

Typically, the AutoSense II™ is mounted between 6 and 7 meters (19.5 and 23 feet), centered above the traffic lane, as shown in Figure 3-2. The AutoSense II™ has mounting holes located at each corner of the base. Mounting bolts are supplied with each unit. A mounting plate is also available that allows the sensor to be mounted to horizontal poles with a diameter from 2 to 3.5 inches (50 to 90 mm). The maximum mounting height is 7.6 meters (25.0 feet). Please reference APPENDIX E for nominal installation information.

3.2.1 Look Down Angle

The AutoSense II will meet the specifications as defined herein when mounted with the correct look down angle. The recommended look down angle is 10 degrees for the first beam and 0 degrees for the second beam. These beam angles are achieved by mounting the sensor with a 5-degree forward tilt as shown in Figure 3-2.

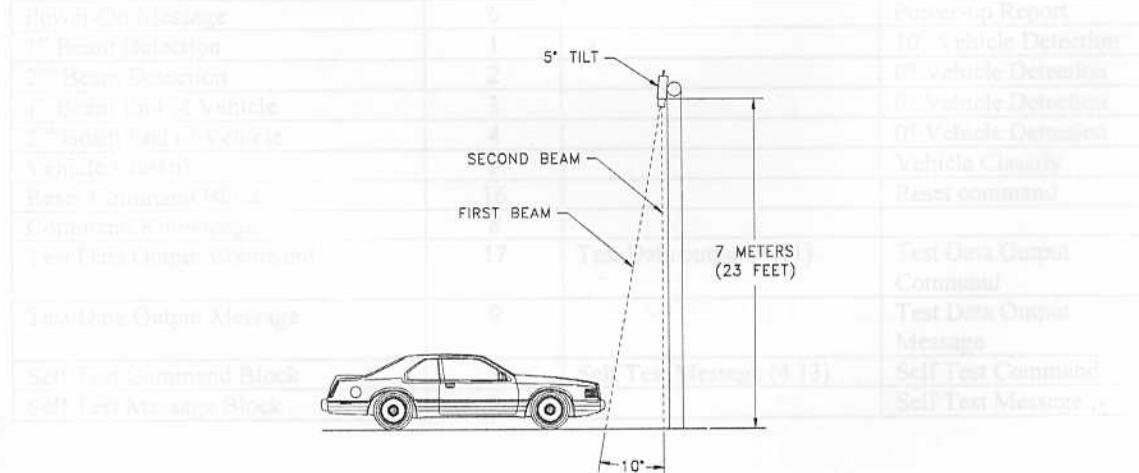


Figure 3-2. Typical AutoSense II™ Mounting.

4.2 POWER-ON MESSAGE**SECTION 4****MESSAGE PROTOCOLS OF THE
AUTOSENSE II™****4.1 DATA FRAME FORMAT.**

For normal operation, the AutoSense II™ is configured for 57.6 Kbaud, no parity, 8 data bits, and 1 stop bit. No hardware or software handshaking is used. The AutoSense II™ will send messages as it detects vehicles. Each data frame will consist of a 2 byte synchronization header, followed by the message/command, and ending with a 1 byte checksum, as shown in the following table. The sections that follow describe the message blocks for each of the AutoSense II™ messages. All 16-bit data is transferred low byte first.

Name	Descriptions	Size	Value
Frame Sync	First byte of a two byte synchronization header.	1 Byte	A5 hex
Frame Start	Second byte of a two byte synchronization header.	1 Byte	5A hex
Message Block	The message/command data will be inserted into this location.	1 – N bytes	See 4.2 through 4.13
Checksum	Byte-wise exclusive-OR of Message block	1 Byte	0 to FF hex

4.2 MESSAGE SUMMARY

Command	Message ID	Response	Description
Power-On Message	6		Power-up Report
1 st Beam Detection	1		10° Vehicle Detection
2 nd Beam Detection	2		0° Vehicle Detection
1 st Beam End of Vehicle	3		0° Vehicle Detection
2 nd Beam End of Vehicle	4		0° Vehicle Detection
Vehicle Classify	5		Vehicle Classify
Reset Command Block	16		Reset command
Command Knowledge	8		
Test Data Output Command	17	Test Data output (4.11)	Test Data Output Command
Test Data Output Message	9		Test Data Output Message
Self Test Command Block	7	Self Test Message (4.13)	Self Test Command
Self Test Message Block	7		Self Test Message

MESSAGE PROTOCOL

4.2.1 POWER-ON MESSAGE BLOCK.

This message will be sent after the AutoSense II™ unit has powered up.

Name	Descriptions	Type	Size	Range/ Value	Precision	Unit of Measure
Message ID	Defines the start of a message	Byte	1	6	N/A	N/A
Self-Test Results	Any faults detected by the internal self-test routines. A set bit indicates a failure. All bits are defined in the Self-Test Message Description	Word	1	0-7FFh	N/A	N/A
Firmware Major Revision	The primary version level of AutoSense II™ Firmware	Byte	1	0-99	1	N/A
Firmware Minor Revision	The secondary version level of the AutoSense II™ Firmware	Byte	1	0-99	1	N/A
Firmware Patch Revision	The patch version level of the AutoSense II™ Firmware	Byte	1	0-99	1	N/A
Range to Road Beam #1	The range from the AutoSense II™ to the road for every degree of the scan for Beam #1	Byte	30	0-255	0.25	¼ Feet
Range to Road Beam #2	The range from the AutoSense II™ to the road for every degree of the scan for Beam #2	Byte	30	0-255	0.25	¼ Feet

4.2.2 FIRST BEAM VEHICLE DETECTION MESSAGE BLOCK: (Message #1)

Name	Descriptions	Type	Size	Range/ Value	Precision	Unit of Measure
Message ID	Defines this as a First Beam Vehicle Detection Message	Byte	1	1	N/A	N/A
Vehicle ID	Unique number assigned by the AutoSense II™ upon detection of a new vehicle	Byte	1	0-255	1	N/A

4.2.3 SECOND BEAM VEHICLE DETECTION MESSAGE BLOCK: (Message #2)

Name	Descriptions	Type	Size	Range/ Value	Precision	Unit of Measure
Message ID	Defines this as a Second Beam Vehicle Detection Message	Byte	1	2	N/A	N/A
Vehicle ID	Unique number assigned by the AutoSense II™ corresponding to Message #1 for the same vehicle.	Byte	1	0-255	1	N/A
Vehicle Left	Defines where the left edge (with respect to the sensor) of the vehicle is positioned	Byte	1	0-29	1	Degrees

MESSAGE PROTOCOL

Edge Position	within the lane					
Vehicle Right Edge Position	Defines where the right edge (with respect to the sensor) of the vehicle is positioned within the lane	Byte	1	0-29	1	Degrees
Vehicle Speed	Speed at the leading edge of the vehicle	Byte	1	0-255	1	MPH

4.2.4 FIRST BEAM END OF VEHICLE DETECTION MESSAGE BLOCK: (Message #3)

Name	Descriptions	Type	Size	Range/Value	Precision	Unit of Measure
Message ID	Defines this as a First Beam End of Vehicle Detection Message	Byte	1	3	N/A	N/A
Vehicle ID	Unique number assigned by the AutoSense II™ corresponding to Messages #1 and #2 for the same vehicle	Byte	1	0-255	1	N/A
Vehicle Left Edge Position	Defines where the left edge (with respect to the sensor) of the vehicle is positioned within the lane	Byte	1	0-29	1	Degrees
Vehicle Right Edge Position	Defines where the right edge (with respect to the sensor) of the vehicle is positioned within the lane	Byte	1	0-29	1	Degrees

4.2.5 SECOND BEAM END OF VEHICLE DETECTION MESSAGE BLOCK: (Message #4)

Name	Descriptions	Type	Size	Range/Value	Precision	Unit of Measure
Message ID	Defines this as a Second Beam End of Vehicle Message	Byte	1	4	N/A	N/A
Vehicle ID	Unique number assigned by the AutoSense II™ corresponding to Messages #1, #2, and #3 for the same vehicle	Byte	1	0-255	1	N/A

4.2.6 VEHICLE CLASSIFICATION MESSAGE BLOCK: (Message #5)

Name	Descriptions	Type	Size	Range/Value	Precision	Unit of Measure
Message ID	Defines this as a Vehicle Classification Message	Byte	1	5	N/A	N/A
Vehicle ID	Unique number assigned by the AutoSense II™ corresponding to previous messages for the same vehicle	Byte	1	0-255	1	N/A
Vehicle Classification	A number which represents the class of a vehicle	Byte	1	0-11	1	N/A
	0 = Unknown 1 = Motorcycle					

MESSAGE PROTOCOL

Name	Descriptions	Type	Size	Range/Value	Precision	Unit of Measure
Classification	2 = Motorcycle with trailer 3 = Passenger Car 4 = Passenger Car w/trailer 5 = Pickup/Van/Sport Utility 6 = Class 5 w/trailer 7 = Single Unit Truck/Bus 8 = Class 7 w/trailer 9 = Tractor w/ 1 trailer 10 = Tractor w/2 trailers 11 = Tractor w/3 trailers					
Classification Confidence	A number which represents the probability that the Vehicle Classification is accurate	Byte	1	0-100	1	Percentage
Feature Data	Vehicle Height	Byte	1	0-255	0.25	Feet
Feature Data	Vehicle Length	Byte	1	0-255	0.25	Feet
Feature Data	Vehicle Width	Byte	1	0-255	1	Degree
Feature Data	Spare	Byte	5	0	N/A	N/A
Vehicle Speed	Speed at leading edge	Byte	1	0-255	1	MPH

4.2.7 RESET COMMAND BLOCK.

This command can be sent by the roadside computer to reset the AutoSense II™ unit.

Name	Descriptions	Type	Size	Range/Value	Precision	Unit of Measure
Command ID	Defines this as a Reset command	Byte	1	16	N/A	N/A

4.2.8 COMMAND ACKNOWLEDGED BLOCK.

This message will be sent in response to the AutoSense II™ unit receiving a Reset Command (4.2.7).

Name	Descriptions	Type	Size	Range/Value	Precision	Unit of Measure
Message ID	Defines the start of a message	Byte	1	8	N/A	N/A

4.2.9 TEST DATA OUTPUT COMMAND BLOCK.

This command can be sent by the roadside computer to receive range and intensity data from the AutoSense II™ unit. The Test Data Output Message (4.2.10) will be sent in response to this command.

Name	Descriptions	Type	Size	Range/ Value	Precision	Unit of Measure
Command ID	Defines this as a Test Data Output Command	Byte	1	17	N/A	N/A

4.2.10 TEST DATA OUTPUT MESSAGE BLOCK.

This message will be sent after the AutoSense II™ unit has received the Test Data Output Command (4.2.9).

Name	Descriptions	Type	Size	Range/ Value	Precision	Unit of Measure
Message ID	Defines the start of a message	Byte	1	9	N/A	N/A
Range to Road Beam #1	The range to the road for every degree of the scan for Beam #1	Byte	30	0-255	0.25	Feet
Intensity of Road Beam #1	The intensity (signal strength) for every degree of the scan for Beam #1	Byte	30	0-127	1	N/A
Range to Road Beam #2	The range to the road for every degree of the scan for Beam #2	Byte	30	0-255	0.25	Feet
Intensity of Road Beam #2	The intensity (signal strength) for every degree of the scan for Beam #2	Byte	30	0-127	1	N/A

4.2.11 SELF-TEST COMMAND BLOCK.

This command can be sent by the roadside computer to initiate a self-test by the AutoSense II™. The Self-Test Message (4.2.12) will be sent in response to this command.

Name	Descriptions	Type	Size	Range/ Value	Precision	Unit of Measure
Command ID	Defines this as a Self-Test Command	Byte	1	7	N/A	N/A

4.2.12 SELF-TEST MESSAGE BLOCK.

This message will be sent in response to receiving a Self-Test Command (4.2.11) or when the AutoSense II™ detects a failure.

Name	Descriptions	Type	Size	Range/ Value	Precision	Unit of Measure
Message ID	Defines the start of a message	Byte	1	7	N/A	N/A
Faults	Any faults detected by the internal self-test routines. A set bit indicates a failure. See SELF-TEST RESULTS FLAGS (section 4.2.12.1) for a description of each item.	Word	1	0-7FFh	N/A	N/A
	Bit 0 = EPROM Checksum Bit 1 = EEPROM Checksum Bit 2 = Internal RAM Bit 3 = External RAM Bit 4 = Motor Control Bit 5 = 5-Volt Supply Bit 6 = APD Temperature Bit 7 = Air Temperature Bit 8 = Calibration Bit 9 = Threshold Bit 10 = High Voltage Bit 11 = T0 Warn Bit 12 = N/A Bit 13 = N/A Bit 14 = N/A Bit 15 = N/A					

4.2.12.1 SELF TEST RESULT FLAGS

Bit	Name	Description
0	EPROM CKSUM	EPROM Checksum. This test should not fail. If so it indicates a corrupted or failing EEPROM.
1	EEPROM CKSUM	Parameter Settings EEPROM Checksum. This test will fail if the checksum is not updated after a EEPROM parameter has been changed.
2	Internal RAM	Microprocessor Internal RAM Read/Write test.
3	External RAM	External SRAM Read/Write test.
4	Motor Control	Indicates the
5	5V POWER	RPE 5 Volt Power supply out of range.
6	APD Temperature	Laser receiver APD temperature reading out of acceptable limits. Possibly defective temperature sensor.
7	Air Temperature	The window temperature sensor value is not within the acceptable limits (0.02 to 3.00 volts.)
8	Calibration	Indicates the laser range calibration numbers are out of acceptable limits.
9	THRESHOLD	Laser receiver threshold setting test. A failure indicates a problem in the HVC controller electronics.

MESSAGE PROTOCOL

10	High Voltage	Laser receiver APD High voltage gain setting out of range. A failure indicates a problem in the HVC controller electronics.
11	T0 warning	Indicates that ten percent or more range readings were less than the minimum range threshold during a three hour period..
12	-	Not used.
13	-	Not used.
14	-	Not used.
15	-	Not used.

Anexo D*Datasheet del Módulo AvyData E711-800*



AnyDATA
AnyTime AnyPlace Any Wireless
Data Solutions

EMII-800 User Manual

Anexo B

Datasheet del Módulo AnyData EMII-800

EMII-800-V2.0

December 5, 2002

AnyDATA.NET Inc.

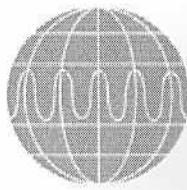
Haenam-Band 303-67

Bwelyeong-dong Kwangju

KOREA

Tel) 82-3-504-3361

Fax) 82-3-504-3361



AnyDATA

AnyTime AnyPlace Any Wireless
Data Solutions

Hanyit Bank B/D 6F
Byulyang-dong Kwachon
KOREA

Copyright © 2001 AnyDATA.NET Inc.
All rights reserved. Printed in the Republic of Korea.

EMII-800 User Manual

Application Information

The information contained in or related to this manual is the property of AnyDATA.NET Inc., and all rights are reserved. Any user agrees that this material and the information contained herein is held in confidence, is in trust and will not be used, copied, reproduced or disclosed to any third party without the express written permission of AnyDATA.NET Inc.

AnyDATA.NET Proprietary Restricted Distribution: This document contains information about AnyDATA.NET products and may not be distributed to others without permission of AnyDATA.NET Inc. All data and information contained in this manual is confidential and copyrighted material of AnyDATA.NET Inc. No part of this document may be reproduced, in any form, without the written permission of AnyDATA.NET Inc.

EMII-800-V2.0

December 5, 2002

Author: The information in this document has been carefully prepared by AnyDATA.NET Inc. does not assume any liability arising out of the application of any information contained in this document. It conveys no license under its patent rights or other intellectual property rights.

Send Technical Questions to:
patterson@AnyDATA.NET

AnyDATA.NET Inc.
Hanyit Bank B/D 6F
Byulyang-dong Kwachon
KOREA

Tel) 82-2-504-3360

Fax) 82-2-504-3362

AnyDATA.NET
1500 Hanyit-dong
Seoul, Korea
130-060-1111
Fax: 82-2-504-3362

.....

Introduction

This Manual gives the hardware interface and programming information for the EMII-800 AnyDATA.NET Modem.

If you can connect the EMII-800 to their **AnyDATA.NET Inc.**
Hanvit Bank B/D 6F
Byulyang-dong Kwachon
KOREA

AnyDATA.NET Inc. assumes no responsibility for any damage to the equipment or loss of data resulting from the use of this product. AnyDATA.NET Inc. reserves no responsibility for damage or claims by third parties, which may arise through the use of this product.

Copyright © 2001 **AnyDATA.NET Inc.**.
All rights reserved. Printed in the Republic Of KOREA.

AnyDATA.NET Inc. assumes no responsibility for damage or loss of data as a result of malfunctions or repairs.

All data and information contained in or disclosed by this document are confidential and proprietary information of **AnyDATA.NET Inc.**, and all rights therein are expressly reserved. By accepting this material, the recipient agrees that this material and the information contained therein are held in confidence and in trust and will not be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of **AnyDATA.NET Inc.**.

AnyDATA.NET Proprietary: Restricted Distribution. This document contains critical information about **AnyDATA.NET** products and may not be distributed to anyone without permission of **AnyDATA.NET Inc.**. All data and information contained in this document are proprietary and confidential information of **AnyDATA.NET Inc.**. No part of this document may be reproduced, in any form or any means without written permission of **AnyDATA.NET Inc.**.

Although the information in this document has been carefully reviewed and it's believed to be reliable, **AnyDATA.NET Inc.** does not assume any liability arising out of the application or use of any product described herein. Neither does it convey any license under its patent rights nor rights of others.

Send Technical Questions to:
paiton@AnyDATA.NET

EMII-800 Reference Manual Application Information
EMII-800-V1.0
February 15, 2002

Operating Requirements

*Restricted Distribution
DO NOT COPY*

Introduction

This Manual provides hardware interface and programming information for EMII-800 CDMA Wireless Data Modem. Users can connect the EMII-800 to their PC or Notebook and easily test the wireless communications.

Disclaimer and Limitation of Liability

AnyDATA.NET Inc. assumes no responsibility for any damage or loss resulting from the misuse of its products. AnyDATA.NET Inc. assumes no responsibility for any loss or claims by third parties, which may arise through the use of its products.

AnyDATA.NET Inc. assumes no responsibility for any damage or loss caused by the deletion or loss of data as a result of malfunctions or repairs.

The information disclosed herein is the exclusive property of AnyDATA.NET Inc. and no part of this publication may be reproduced or transmitted in any form or by any means including electronic storage, reproduction, execution or transmission without the prior written consent of AnyDATA.NET Inc. The information contained in this document is subject to change without notice.

Reproduction, adaptation or translation of this document is prohibited without prior written permission of AnyDATA.NET Inc.

FCC RF Exposure Information

Warning! Read this information before using this device.



In August 1996 the Federal Communications Commission (FCC) of the United States with its action in Report and Order FCC 96-326 adopted an updated safety standard for human exposure to radio frequency electromagnetic energy emitted by FCC regulated transmitters. Those guidelines are consistent with the safety standard previously set by both U.S. and international standards bodies. The design of this device complies with the FCC guidelines and these international standards.



CAUTION

- Operating Requirements

restricted Distribution
DO NOT COPY

- The user cannot make any changes or modifications not expressly approved by the party responsible for compliance; otherwise it could void the user's authority to operate the equipment.
- To satisfy FCC RF exposure compliance requirements for a mobile transmitting device, this device and its antenna should generally maintain a separation distance of 20cm or more from a person's body.

1.2 Organization

Special accessories

In order to ensure that this device is in compliance with FCC regulation, special accessories are provided with this device and must be used with this device only. The user should not use accessories other than the special accessories given with this device

2 Overview

- 2.1 Application Description
- 2.2 Technical Specifications
 - 2.2.1 General Specification
 - 2.2.2 Receive Specification
 - 2.2.3 Transmit Specification
 - 2.2.4 Standards
- 2.3 Interface Diagram
- 2.4 EMII-800 General Features
- 2.5 Internal Module Features

3 Power Sources

- 3.1 AC Power Source (RS232 Standard)
- 3.2 9-Volt DC Power Source
- 3.3 DC Power Adapter

4 Interface Descriptions

- 4.1 Overview
- 4.2 RS232 Interface (Standard)
- 4.3 IFR

restricted Distribution
DO NOT COPY

Contents

-
- 2.1 Absolute Maximum Ratings
 - 2.2 Recommended Operating Conditions
 - 1 Introduction**
 - 1.1 Purpose
 - 1.2 Organization
 - 1.3 Revision History
 - 1.4 References
 - 1.5 Acronym List
 - 7 FCC Notices
 - 2 Overview**
 - 2.1 Application Description
 - 2.2 Technical Specifications
 - 2.2.1 General Specification
 - 2.2.2 Receive Specification
 - 2.2.3 Transmit Specification
 - 2.2.4 Standards
 - 2.3 Interface Diagram
 - 2.4 EMII-800 General Features
 - 2.5 Internal Module Features
 - 3 PIN Description**
 - 3.1 8-Pin Connector (RS232 Standard)
 - 3.2 3-Pin Connector (Debugging)
 - 3.3 DC Power Connector
 - 4 Interface Descriptions**
 - 4.1 Overview
 - 4.2 RS232 Interface (Standard)
 - 4.3 LED

restricted distribution
DO NOT COPY

5 Electrical Specifications

- 5.1 Absolute Maximum Ratings
- 5.2 Recommended Operating Conditions
- 5.3 Power Consumption
- 5.4 Serial Interface Electrical Specifications

6 Mechanical Dimension

6.1 EMII-800 Outline

7 FCC Notice

- # Section 1 – Contains EMII-800 Pin description, Pin Function, Pin Port, and Pin Configuration.
- # Section 2 – Describes the I2C™ Interface.
- # Section 3 – Specifies the recommended operating conditions, DC voltage, resistance, DC current, and power estimates for the module.
- # Section 6 – Provides package dimensions and outline drawings for the module.
- # Section 7 – Describes the FCC ID.

1.3 Revision History

The revision history for this document is shown in Table 1-1.

Table 1-1 Revision History

Version	Date	Description
V1.0	July 10, 2001	Initial Release
V1.1	Dec. 2001	Minor revision made to the

1.4 References

- 1. DIA/DO/D Incorporated. Minimum performance specification for Component 3000 Series modules. 90-V2180-7(X), July 10, 2001.
- 2. DIA/DO/D Incorporated. MSMSM8027 module design. Module Design Preliminary Information 90-V2180-1-03, August 10, 2001.
- 3. DIA/DO/D Incorporated. SURE5100 User Manual. 90-V2180-1-X1, 2001.

1. Introduction

1.1 Purpose

This Manual provides hardware interface and programming information for the EMII-800 CDMA Wireless Data Module.

1.2 Organization

This manual discusses the interface and operation of the module and is divided into the following subsections:

- Section 2 – Introduces users to the EMII-800 CDMA Wireless Data Module's basic features and general specifications.
- Section 3 – Contains EMII-800 Pin descriptions - DC12V Input Port, 8-pin Serial Port, and Debugging Port.
- Section 4 – Describes the UART Interface.
- Section 5 – Specifies the recommended operating conditions, DC voltage characteristics, I/O timing, and power estimations for the module.
- Section 6 – Provides package dimensions and outlook features for the module.
- Section 7 – Describes the FCC Notice.

1.3 Revision History

The revision history for this document is shown in Table 1-1.

Table 1-1 Revision History

Version	Date	Description
V1.0	Feb. 2002	Initial Release
V2.0	Dec. 2002	Corrected document content

1.4 References

1. QUALCOMM Incorporated. MSM5100 Mobile Station Modem™: Component Supply Specification. 80-V2180-7-X1, July 13, 2001.
2. QUALCOMM Incorporated. MSM5100™ Mobile Station Modem: Device Specification (Preliminary Information). 93-V2180-1-X3, August 30, 2001.
3. QUALCOMM Incorporated. SURF5100 User Manual. 80-V2535-1-X1, March 28, 2001.

1.5 Acronym List

Term	Definition
CDMA	Code-Division Multiple Access
CODEC	Coder-Decoder
GPIO	General-purpose Input/Output
JTAG	Joint Test Action Group (ANSI/IEEE Std. 1149.1-1990)
LCD	Liquid Crystal Display
LDO	Voltage Regulator
LED	Light Emitting Diode
PCB	Printed Circuit Board
PCM	Pulse Coded Modulation
PCS	Personal Communications Service
RF	Radio Frequency
Rx	Receive
TCXO	Temperature-Controlled Crystal Oscillator
Tx	Transmit
UART	Universal Asynchronous Receiver Transmitter

2. Overview

2.1 Application Descriptions

The CDMA Wireless Data Module is a complex consumer communications instrument that relies heavily on both digital signal and embedded processor technologies. The Wireless Data Modules manufactured by AnyDATA.NET support Code-Division-Multiple-Access (CDMA). This operates in the PCS spectrum.

In a continuing effort to simplify the design and to reduce the production cost of the Wireless Data Module, AnyDATA.NET has successfully developed the EMII-800. The EMII-800 is AnyDATA.NET's latest compact Wireless Data Module operating in the PCS spectrum. The EMII-800 contains a complete digital modulation and demodulation system for CDMA standards as specified in IS-95 A/B and IS-2000.

The subsystem within the EMII-800 includes a CDMA processor (MSM5100), an integrated CODEC with an ear piece and microphone amplifiers, and an RS-232 serial interface supporting forward link data communications at a rate of 153.6kbps.

The EMII-800 provides an external interface, that includes the standard RS-232 and Digital Audio.

The EMII-800 has the capability to power down unused circuits in order to dynamically minimize power consumption.

Frequency Range		Modulation	
Antenna	Wing Antenna, omnian	Antenna	Wing Antenna, omnian
2.4 GHz	802.11b IEEE 802.11b	2.4 GHz	802.11b IEEE 802.11b
5.8 GHz	802.11a IEEE 802.11a	5.8 GHz	802.11a IEEE 802.11a
800-900 MHz	GSM/GPRS	800-900 MHz	GSM/GPRS
1800-1900 MHz	DCS/PCS	1800-1900 MHz	DCS/PCS

restricted Distribution
DO NOT COPY

2.2 Technical Specifications

2.2.1 General Specifications

PARAMETERS	DESCRIPTIONS
External Access	Code-Division-Multiple-Access (CDMA)
CDMA Protocol	IS-95A/B/C, IS-98A, IS-126, IS-637A, IS-707A
Data Rate	153.6Kbps
Transmit/Receive Frequency Interval	45MHz
Band Width	1.23MHz
Operating Voltage	DC 6V ~ 12V
Current Consumption	Stand by mode: Idle (55mA), Busy mode: 280mA (Max) at 12V
Operating Temperature	-30°C ~ +60°C
Frequency Stability	±300Hz
Antenna	Whip Antenna, 50ohm
Size	57 X 121 X 24mm with case
Weight	About 110g
External Interface	RS-232, Digital/Analog Audio, LCD, Keypad, Ringer

IS-95A/B/C: Direct Sequence Spread Spectrum
IS-98A: Voice Recognition
IS-126: Direct Sequence Spread Spectrum
IS-637: Voice Loop-Back
IS-707: Short Message Service
IS-707: Data Service

Restricted Distribution
DO NOT COPY

2.2.2 Receive Specifications

PARAMETERS	DESCRIPTIONS
Frequency Range	869.04 ~ 893.97 MHz
Sensitivity	Below -104 dBm
Interference Rejection	Single tone (-30dBm @900KHz): Below -101dBm Two tone (-43 dBm @900KHz and 1700KHz): Below -101dBm Two tone (-32 dBm @900KHz and 1700KHz): Below -90dBm Two tone (-21 dBm @900KHz and 1700KHz): Below -79dBm
Spurious Wave Suppression	Below -80dBc
Input Dynamic Range	-25 dBm ~ -104dBm

2.2.3 Transmit Specifications

PARAMETERS	DESCRIPTIONS
Frequency Range	824.04 ~ 848.97 MHz
Nominal Power	0.32 W
Minimum Controlled Output Power	Below -50dBm
Max Power Spurious	900KHz: Below -42dBc/30KHz 1.98MHz: Below -54dBc/30KHz

2.2.4 Standards

IS-95A/B/C: Protocol Between MS & BTS
 IS-96A: Voice Signal Coding
 IS-98A: Base MS Function
 IS-126: Voice Loop-Back
 IS-637: Short Message Service
 IS-707: Data Service

Restricted Distribution
DO NOT COPY

2.3 Interface Diagram

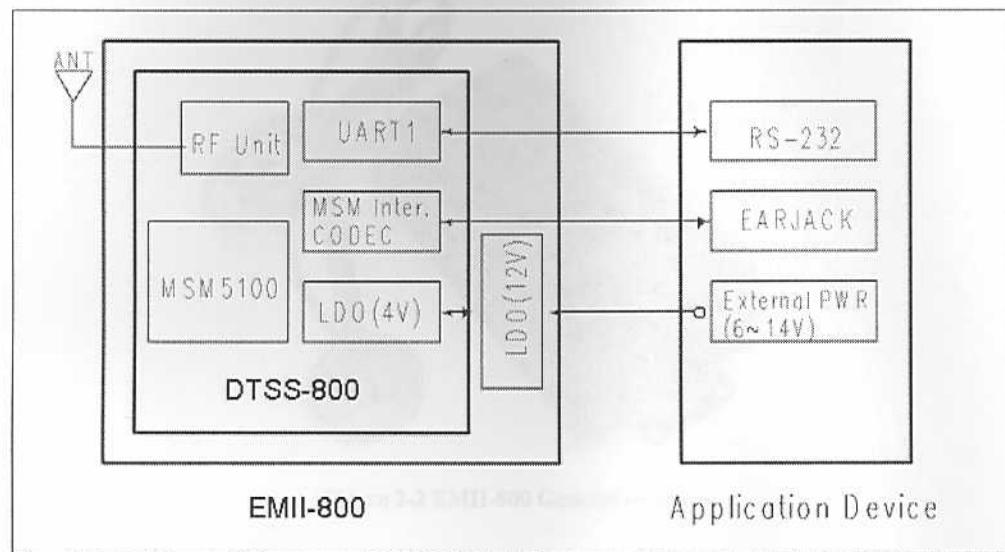


Figure 2-1 Interface Block Diagram

Figure 2-2 Internal Module General Layout (Top View)

restricted Distribution
DO NOT COPY

2.4 EMII-800 General Features



Figure 2-2 EMII-800 General Features

2.5 Internal Module Features

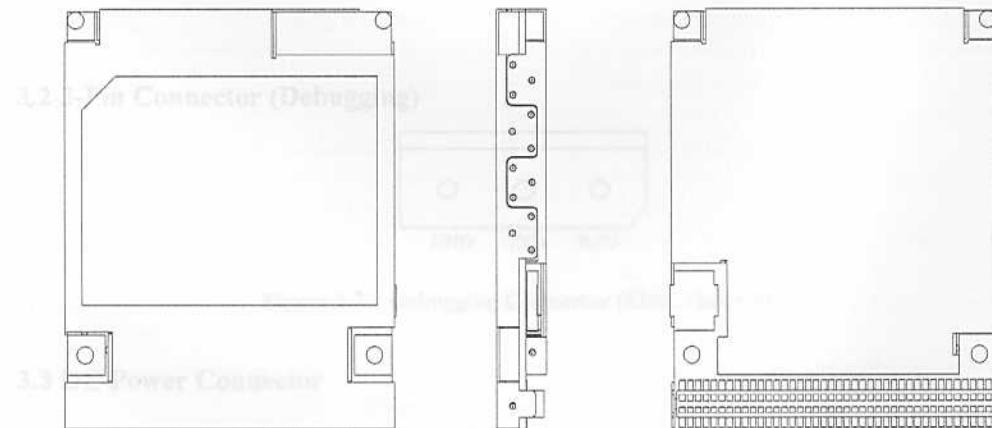


Figure 2-3 Internal Module General Features (DTSS-800)

RESTRICTED DISTRIBUTION
DO NOT COPY

3. PIN Description

3.1 8-Pin Male Modular Jacks (RS232 Standard)

Molex (85503)	RS-232
Pin 1	RI
Pin 2	GND
Pin 3	DTR
Pin 4	CTS
Pin 5	TX
Pin 6	RTS
Pin 7	RX
Pin 8	DCD

Figure 3-1 Right Angle Modular Jacks Pin Description (85503,Molex 8P)

3.2 3-Pin Connector (Debugging)

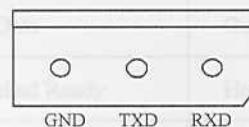


Figure 3-2 Debugging Connector (5268, Molex 3P)

3.3 DC Power Connector

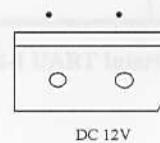


Figure 3-3 DC 12V Power Connector (5268, Molex 2P)

Restricted Distribution
DO NOT COPY

4. Interface Descriptions

4.1 Overview

This chapter provides essential information that the user needs to understand, in order to convert the EMII-800 into a subscriber unit application. In addition, the internal signals that are necessary for the complete understanding of the UART interfaces are described below.

4.2 RS232 Interface (Standard)

The Universal Asynchronous Receiver Transmitter (UART) communicates with serial data that conforms to the RS-232 Interface protocol. The module inside the EMII-800 provides 3.0V CMOS level outputs and 3.0V CMOS switching input levels. All inputs have a 5.0V tolerance, but 3.0V or 3.3V CMOS logic compatible signals are highly recommended.

All the control signals of the RS-232 are active low, but the data signals, RXD and TXD, are active high. The UART has a 64 byte transmit (TX) FIFO and a 64 byte receive (RX) FIFO. The UART features hardware handshaking, programmable data sizes, programmable stop bits, and odd, even, no parity. The UART operates at a maximum bit rate of 115.2kbps.

NAME	DESCRIPTION	CHARACTERISTIC
DP_DCD/	Data Carrier Detect	Network connected from the module
DP_RI/	Ring Indicator	Output to host indicating coming call
DP_RTS/	Request to Send	Ready for receive from host
DP_TXD	Transmit Data	Output data from the module
DP_DTR/	Data Terminal Ready	Host ready signal
DP_RXD	Receive Data	Input data to the module
DP_CTS/	Clear to Send	Module output signal
GND	Signal Ground	Signal ground

Figure 4-1 UART Interface Pinouts

4.3 LEDs

The EMII-800 has four LEDs that indicate the status and functionality of the module inside. The PWR LED turns on when adequate power is supplied to the module, and the module is able to turn on. The SMS LED will turn on, if there is a SMS message or a voicemail message. After the user has read the SMS message or listened to the voicemail message, the SMS LED will turn off. When the module is in the traffic or conversation stage, the BUSY LED will be on. Shortly after the module has been turned on, the IDLE LED should turn on indicating that the module is in-service. This means that the module is within the range of the base station and is able to receive a signal from the base station.

5. Electrical Specifications

5.1 Absolute Maximum Ratings

Operating the module under conditions that exceed those listed in the Absolute Maximum Ratings table may result in damage to the module.

Absolute Maximum Ratings should be considered as limiting value. The module may not function properly and should not be operated if any one of the parameters is not within its specified operating range.

Table 5-1 Absolute Maximum Ratings

PARAMETER	MIN	MAX	UNITS
Storage Temperature	-40	+80	°C
Voltage On Any Input Pin	-	+20.0	V
Voltage On Any Output Pin	-	+10.0	V
Supply Voltage	-	+15	V
Initializing Current	100		mA
Drop	No damages after 60-Inch drop over concrete floor		

5.2 Recommended Operating Conditions

PARAMETER	MIN	MAX	UNITS
Supply Voltage	+6	+12	V
Operating Temperature	-30	+60	°C
Operating Humidity	95%(50°C) Relative Humidity		

5.3 Power Consumption

CONVERSATION	STANDBY	
	Idle	Sleep
280 mA (MAX)	55 mA	18 mA

Note: Power consumption measured with a supply voltage of 12V

5.4 Serial Interface Electrical Specifications

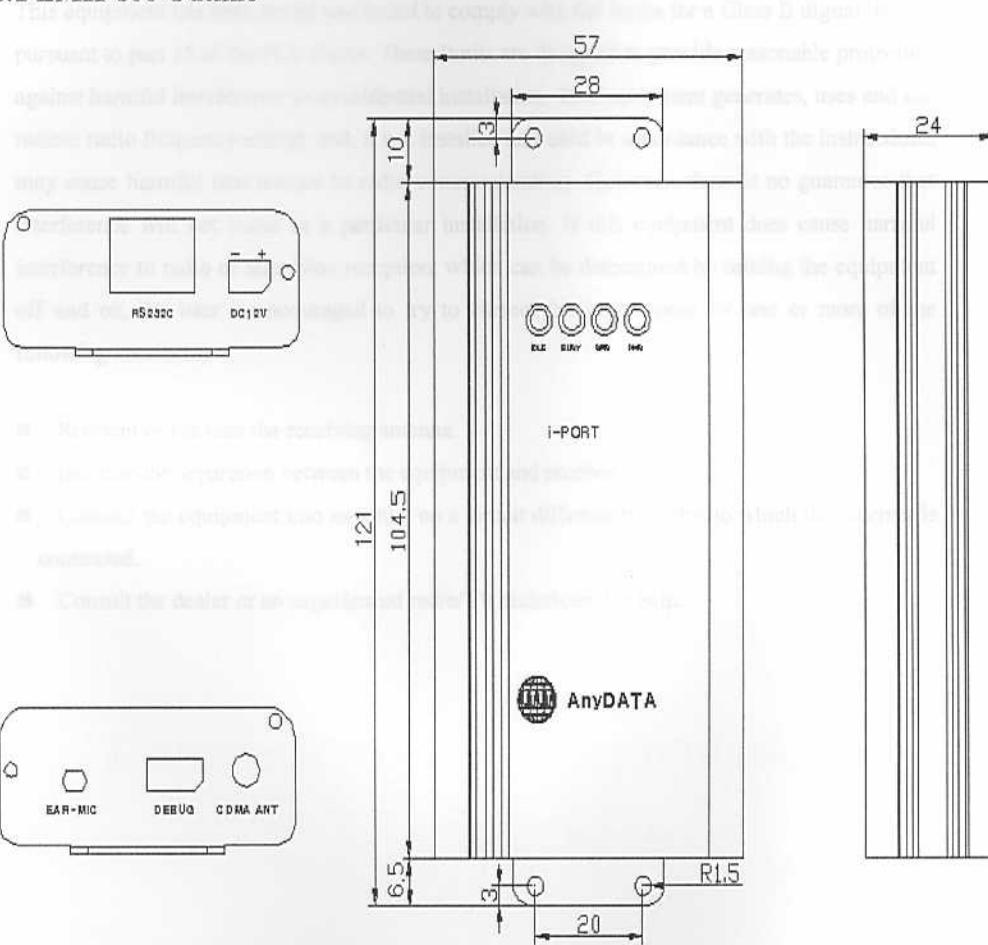
PARAMETER	MIN	TYP	MAX	UNITS
Input High Voltage	+1.7		+15.0	V
Input Low Voltage	-15.0		+1.2	V
Output High Voltage	+5.0	+6.0	+7.0	V
Output Low Voltage	-7.0	-6.0	-5.0	V

.....

Restricted Distribution
DO NOT COPY

6. Mechanical Dimensions

6.1 EMII-800 Outline



Units: mm

Restricted Distribution
DO NOT COPY

7. FCC Notice

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

*Restricted Distribution
DO NOT COPY*

Architecture	High-performance 16-bit RISC architecture
Clock Sources	- Processor clock up to 12 MHz - Internal oscillator - External oscillator input - PLL with 100% digital control - Frequency synthesis with 100% digital control - 16-bit programmable timer
Memory	- 8KB bytes Flash (user) - 1KB 16 Flash (reserved) - 1KB 16 ROM (reserved) - External parallel data memory
External Peripherals	- Frequency up to 3.75 MHz - Serial port with 120 TM baudrate - SPI interface compatibility - 10-bit ADC with 100ns conversion time - 10-bit DAC with 100ns conversion time - 10-bit DAC with 100ns conversion time
On-Chip JTAG Debug & Download	- On-chip debug interface for software development and in-circuit debugging - On-chip flash eraser, programmer and debugger - On-chip memory monitor and memory dump - On-chip timer and counter - On-chip timer and counter

Anexo C

Datasheet del Microcontrolador Silabs

C8051F120

Analog Peripherals

12-Bit ADC

- ± 1 LSB INL; no missing codes
- Programmable throughput up to 100 ksp/s
- 8 external inputs; programmable as single-ended or differential
- Programmable amplifier gain: 16, 8, 4, 2, 1, 0.5
- Data-dependent windowed interrupt generator
- Built-in temperature sensor (± 3 °C)

8-Bit ADC

- ± 1 LSB INL; no missing codes
- Programmable throughput up to 500 ksp/s
- 8 external inputs
- Programmable amplifier gain: 4, 2, 1, 0.5

Two 12-Bit DACs

- Can synchronize outputs to timers for jitter-free waveform generation

Two Comparators

Internal Voltage Reference

V_{DD} Monitor/Brown-out Detector

On-Chip JTAG Debug & Boundary Scan

- On-chip debug circuitry facilitates full speed, non-intrusive in-system debug (no emulator required)
- Provides breakpoints, single stepping, watchpoints, stack monitor
- Inspect/modify memory and registers
- Superior performance to emulation systems using ICE-chips, target pods, and sockets
- IEEE1149.1 compliant boundary scan

High-Speed 8051 µC Core

- Pipelined instruction architecture; executes 70% of instructions in 1 or 2 system clocks
- Up to 100 MIPS throughput with 100 MHz system clock
- 16 x 16 multiply/accumulate engine (2-cycle)

Memory

- 8448 bytes data RAM
- 128 kB Flash; in-system programmable in 1024-byte sectors (1024 bytes are reserved)
- External parallel data memory interface

Digital Peripherals

- 64 port I/O; all are 5 V tolerant
- Hardware SMBus™ (I2C™ compatible), SPI™, and two UART serial ports available concurrently
- Programmable 16-bit counter/timer array with six capture/compare modules
- 5 general-purpose 16-bit counter/timers
- Dedicated watchdog timer; bidirectional reset
- Real-time clock mode using timer 3 or PCA

Clock Sources

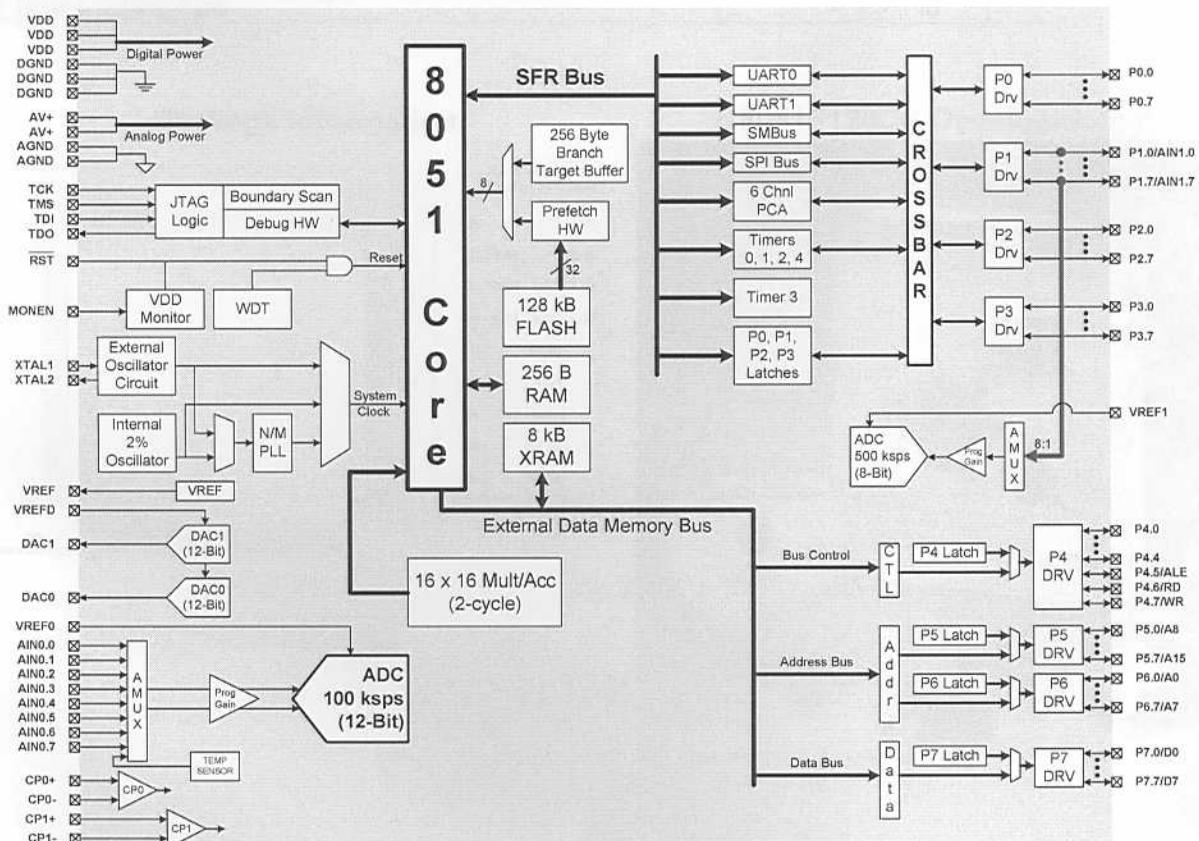
- Internal oscillator: 24.5 MHz, 2% accuracy supports UART operation
- On-chip programmable PLL: up to 100 MHz
- External oscillator: Crystal, RC, C, or Clock

Supply Voltage: 3.0 to 3.6 V

- Typical operating current: 50 mA at 100 MHz
- Typical stop mode current: 0.4 uA

100-Pin TQFP

Temperature Range: -40 to +85 °C

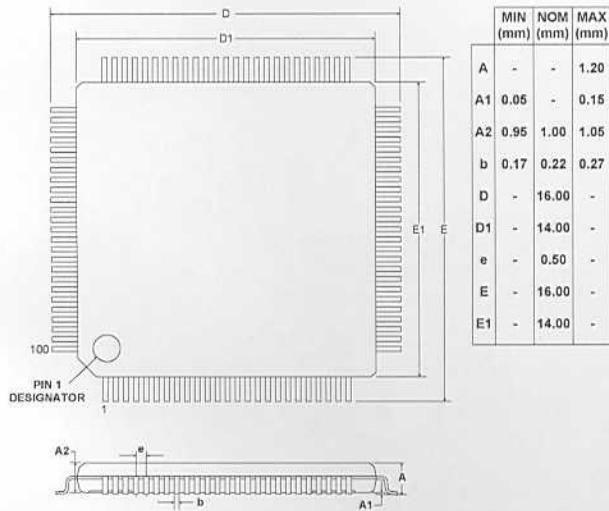


**C8051F120**

100 MIPS, 128 kB Flash, 12-Bit ADC, 100-Pin Mixed-Signal MCU

Selected Electrical Specifications(T_A = -40 to +85 °C, V_{DD} = 3.0 V unless otherwise specified)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
GLOBAL CHARACTERISTICS					
Supply Voltage		3.0		3.6	V
Supply Current (CPU active)	Clock = 100 MHz Clock = 1 MHz Clock = 32 kHz		50 0.6 16		mA mA μA
Supply Current (shutdown)	Oscillator off; V _{DD} Monitor Enabled Oscillator off; V _{DD} Monitor Disabled		10 0.4		μA μA
Clock Frequency Range		DC		100	MHz
INTERNAL CLOCKS					
Oscillator Frequency		24.0	24.5	25.0	MHz
PLL Frequency		96	98	100	MHz
A/D CONVERTER					
Resolution			12		bits
Integral Nonlinearity				±1	LSB
Differential Nonlinearity	Guaranteed Monotonic			±1	LSB
Signal-to-Noise Plus Distortion		66	69		dB
Throughput Rate				100	ksp/s
D/A CONVERTERS					
Resolution			12		bits
Differential Nonlinearity	Guaranteed Monotonic			±1	LSB
Output Settling Time			10		μs

Package Information**C8051F120DK Development Kit**