

# AutomaticElectronicTollCollection

Generated by Doxygen 1.6.3

Tue Nov 9 22:53:59 2010



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	src/AETC_GSMF.c File Reference . . . . .	3
2.1.1	Detailed Description . . . . .	3
2.1.2	LICENSE . . . . .	4
2.1.3	Function Documentation . . . . .	4
2.1.3.1	del_2s . . . . .	4
2.1.3.2	port_close . . . . .	4
2.1.3.3	port_config . . . . .	4
2.1.3.4	port_open . . . . .	4
2.1.3.5	write_data . . . . .	4
2.2	src/AETC_GSMS.c File Reference . . . . .	6
2.2.1	Detailed Description . . . . .	6
2.2.2	LICENSE . . . . .	6
2.2.3	Function Documentation . . . . .	7
2.2.3.1	del_2s . . . . .	7
2.2.3.2	main . . . . .	7
2.2.3.3	port_close . . . . .	7
2.2.3.4	port_config . . . . .	7
2.2.3.5	port_open . . . . .	7
2.2.3.6	write_data . . . . .	7
2.3	src/AETC_lcd.c File Reference . . . . .	8
2.3.1	Detailed Description . . . . .	9
2.3.2	LICENSE . . . . .	9
2.3.3	Function Documentation . . . . .	9
2.3.3.1	lcd_cursor . . . . .	9

2.3.3.2	lcd_init . . . . .	9
2.3.3.3	lcd_print . . . . .	9
2.3.3.4	lcd_set_4bit . . . . .	10
2.3.3.5	lcd_string . . . . .	10
2.3.3.6	lcd_wr_char . . . . .	10
2.3.3.7	lcd_wr_command . . . . .	10
2.4	src/AETC_Robot.c File Reference . . . . .	11
2.4.1	Detailed Description . . . . .	12
2.4.2	LICENSE . . . . .	12
2.4.3	Function Documentation . . . . .	13
2.4.3.1	ADC_Conversion . . . . .	13
2.4.3.2	adc_init . . . . .	13
2.4.3.3	init_devices . . . . .	13
2.4.3.4	main . . . . .	13
2.4.3.5	motion_set . . . . .	14
2.4.3.6	port_init . . . . .	14
2.4.3.7	print_sensor . . . . .	14
2.4.3.8	SIGNAL . . . . .	14
2.4.3.9	timer1_init . . . . .	14
2.4.3.10	timer5_init . . . . .	15
2.4.3.11	uart0_init . . . . .	15
2.4.3.12	velocity . . . . .	15
2.4.4	Variable Documentation . . . . .	16
2.4.4.1	ADC_Value . . . . .	16

# Chapter 1

## File Index

### 1.1 File List

Here is a list of all documented files with brief descriptions:

src/ <a href="#">AETC_GSMF.c</a> . . . . .	3
src/ <a href="#">AETC_GSMS.c</a> . . . . .	6
src/ <a href="#">AETC_lcd.c</a> . . . . .	8
src/ <a href="#">AETC_Robot.c</a> . . . . .	11



## Chapter 2

# File Documentation

### 2.1 src/AETC\_GSMF.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <malloc.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <termios.h>
#include <sys/time.h>
#include <sys/types.h>
```

#### Functions

- int [port\\_open](#) (void)
- void [port\\_config](#) (int fd)
- void [del\\_2s](#) (void)
- void [write\\_data](#) (int fd, char \*c)
- void [port\\_close](#) (int fd)
- int **main** (void)

#### 2.1.1 Detailed Description

Program for serial communication between Owner and TollPlaza through GSM Module. Send Fail Message

#### Author

Puskar Kothavade, Ashish Pardhi, Mugdha Nazare, IIT Bombay

#### Date

10/Oct/2010

**Version**

1.0

**2.1.2 LICENSE**

Copyright (c) 2010. ERTS Lab IIT Bombay All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holders nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Source code can be used for academic purpose. For commercial use permission form the author needs to be taken.

**2.1.3 Function Documentation****2.1.3.1 void del\_2s (void)**

Generate Delay of two seconds.

**2.1.3.2 void port\_close (int *fd*)**

Close port for reuse

**Parameters**

*fd* File Descriptor to access serial port

**2.1.3.3 void port\_config (int *fd*)**

Configure port

**Parameters**

*fd* File Descriptor to access serial port

**2.1.3.4 int port\_open (void)**

Open port for serial communication. display error message if unable to open.

**2.1.3.5 void write\_data (int *fd*, char \* *c*)**

Send Message to register owner



### Parameters

*fd* File Descriptor to access serial port

*\*c* character pointer

## 2.2 src/AETC\_GSMS.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <malloc.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <termios.h>
#include <sys/time.h>
#include <sys/types.h>
```

### Functions

- int [port\\_open](#) (void)
- void [port\\_config](#) (int fd)
- void [del\\_2s](#) (void)
- void [write\\_data](#) (int fd, char \*c)
- void [port\\_close](#) (int fd)
- int [main](#) (void)

### 2.2.1 Detailed Description

Program for serial communication between Owner and TollPlaza through GSM Module. Send Success Message

#### Author

Puskar Kothavade, Ashish Pardhi, Mugdha Nazare, IIT Bombay

#### Date

10/Oct/2010

#### Version

1.0

### 2.2.2 LICENSE

Copyright (c) 2010. ERTS Lab IIT Bombay All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holders nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Source code can be used for academic purpose. For commercial use permission form the author needs to be taken.

### 2.2.3 Function Documentation

#### 2.2.3.1 void del\_2s (void)

Generate Delay of two seconds.

#### 2.2.3.2 int main (void)

< File Descriptor to access serial port

#### 2.2.3.3 void port\_close (int *fd*)

Close port for reuse

##### Parameters

*fd* File Descriptor to access serial port

#### 2.2.3.4 void port\_config (int *fd*)

Configure port

##### Parameters

*fd* File Descriptor to access serial port

#### 2.2.3.5 int port\_open (void)

Open port for serial communication. display error message if unable to open.

#### 2.2.3.6 void write\_data (int *fd*, char \* *c*)

Send Message to register owner

##### Parameters

*fd* File Descriptor to access serial port

\**c* character pointer

## 2.3 src/AETC\_lcd.c File Reference

```
#include <avr/io.h>
#include <avr/delay.h>
#include <util/delay.h>
```

### Defines

- #define [FCPU](#) 11059200ul  
*set the cpu frequency*
- #define [RS](#) 0  
*set command input*
- #define [RW](#) 1  
*writing to LCD*
- #define [EN](#) 2  
*set Enable pin*
- #define **lcd\_port** PORTC
- #define **sbit**(reg, bit) reg |= (1<<bit)
- #define **cbit**(reg, bit) reg &= ~(1<<bit)

### Functions

- void **init\_ports** ()
- void **lcd\_reset\_4bit** ()
- void [lcd\\_init](#) ()
- void [lcd\\_wr\\_command](#) (unsigned char)
- void [lcd\\_wr\\_char](#) (char)
- void **lcd\_home** ()
- void [lcd\\_cursor](#) (char, char)
- void [lcd\\_print](#) (char, char, unsigned int, int)
- void [lcd\\_string](#) (char \*)
- void [lcd\\_set\\_4bit](#) ()

### Variables

- unsigned int **temp**
- unsigned int **unit**
- unsigned int **tens**
- unsigned int **hundred**
- unsigned int **thousand**
- unsigned int **million**
- int **i**

### 2.3.1 Detailed Description

Program for printing data on LCD.

**Author**

Puskar Kothavade, Ashish Pardhi, Mugdha Nazare, IIT Bombay

**Date**

10/Oct/2010

**Version**

1.0

### 2.3.2 LICENSE

Copyright (c) 2010. ERTS Lab IIT Bombay All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holders nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Source code can be used for academic purpose. For commercial use permission form the author needs to be taken.

### 2.3.3 Function Documentation

#### 2.3.3.1 void lcd\_cursor (char *row*, char *column*)

Position the LCD cursor at "*row*", "*column*".

**Parameters**

*row* no. of rows

*column* no. of columns

#### 2.3.3.2 void lcd\_init ()

Function to Initialize LCD

#### 2.3.3.3 void lcd\_print (char *row*, char *coloumn*, unsigned int *value*, int *digits*)

Function To Print Any input value upto the desired digit on LCD

**2.3.3.4 void lcd\_set\_4bit ()**

Function to Reset LCD

**2.3.3.5 void lcd\_string (char \* *str*)**

Function to Print String on LCD

**2.3.3.6 void lcd\_wr\_char (char *letter*)**

Function to Write Data on LCD

**2.3.3.7 void lcd\_wr\_command (unsigned char *cmd*)**

Function to Write Command on LCD

## 2.4 src/AETC\_Robot.c File Reference

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <avr/signal.h>
#include <math.h>
#include "AETC_lcd.c"
#include <avr/delay.h>
```

### Defines

- #define [FCPU](#) 11059200ul  
*defined here to make sure that program works properly*

### Functions

- void [port\\_init](#) ()
- void [timer5\\_init](#) ()
- void [velocity](#) (unsigned char, unsigned char)
- void [motors\\_delay](#) ()
- void [timer1\\_init](#) ()
- unsigned char [ADC\\_Conversion](#) (unsigned char)
- **ISR** (TIMER1\_OVF\_vect)
- void [lcd\\_port\\_config](#) (void)
- void [adc\\_pin\\_config](#) (void)
- void [motion\\_pin\\_config](#) (void)
- void [adc\\_init](#) ()
- void [print\\_sensor](#) (char row, char coloumn, unsigned char channel)
- void [motion\\_set](#) (unsigned char Direction)
- void [forward](#) (void)
- void [stop](#) (void)
- void [uart0\\_init](#) (void)
- void [init\\_devices](#) (void)
- [SIGNAL](#) (SIG\_USART0\_RECV)
- int [main](#) ()

### Variables

- unsigned char [ADC\\_Value](#)
- unsigned char **flag1** = 0
- unsigned char **flag2** = 0
- unsigned char **f** = 0  
*stores flag values*

- unsigned char `Left_white_line` = 0  
*store left white line sensor value*
- unsigned char `Center_white_line` = 0  
*store center white line sensor value*
- unsigned char `Right_white_line` = 0  
*store right white line sensor value*
- unsigned char `Front_Sharp_Sensor` = 0  
*store front sensor value*
- unsigned char `Front_IR_Sensor` = 0
- unsigned int `vehicle_stop` = 0
- unsigned int `LID_Transmit` = 0  
*store transmitted ID value*
- unsigned char `data`  
*store data*
- float `speed`  
*store speed*
- unsigned char `speed_int`
- unsigned char `speed_dec`
- unsigned int `RegSpeed` = 150
- unsigned char `StartTheBot` = 0

### 2.4.1 Detailed Description

Program for speed calculation, white line follower and obstacle detection.

#### Author

Puskar Kothavade, Ashish Pardhi, Mugdha Nazare, IIT Bombay

#### Date

10/Oct/2010

#### Version

1.0

### 2.4.2 LICENSE

Copyright (c) 2010. ERTS Lab IIT Bombay All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.



Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holders nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Source code can be used for academic purpose. For commercial use permission form the author needs to be taken.

### 2.4.3 Function Documentation

#### 2.4.3.1 unsigned char ADC\_Conversion (unsigned char *Ch*)

- < Set start conversion bit
- < Wait for conversion to complete
- < clear ADIF (ADC Interrupt Flag) by writing 1 to it

#### 2.4.3.2 void adc\_init ()

- < MUX5 = 0
- < Vref=5V external --- ADLAR=1 --- MUX4:0 = 0000
- < ADEN=1 --- ADIE=1 --- ADPS2:0 = 1 1 0

#### 2.4.3.3 void init\_devices (void)

- < Clears the global interrupts
- < initialise the ports

#### 2.4.3.4 int main (void)

Main Function

- < Getting data of Left WL Sensor
- < Getting data of Center WL Sensor
- < Getting data of Right WL Sensor
- < Getting data of Front Sharp sensor
- < Getting data of Front IR sensor
- < Prints value of White Line Sensor1
- < Prints Value of White Line Sensor2
- < Prints Value of White Line Sensor3
- < vehicle at first black patch
- < Timer 1 start with no prescaler
- < Getting data of Left WL Sensor
- < Getting data of Center WL Sensor

- < Getting data of Right WL Sensor
- < vehicle at second black patch
- < Timer 1 stop
- < vehicle at third black patch
- < Capturing photo in progress
- < photograph taken. Vehicle can go ahead now
- < Obstacle detection

#### **2.4.3.5 void motion\_set (unsigned char *Direction*)**

Function used for setting motor's direction.

- < removing upper nibbel for the protection
- < reading the PORTA original status
- < making lower direction nibbel to 0
- < adding lower nibbel for forward command and restoring the PORTA status
- < executing the command

#### **2.4.3.6 void port\_init ()**

Functions prototype

#### **2.4.3.7 void print\_sensor (char *row*, char *coloumn*, unsigned char *channel*)**

Function To Print Sesor Values At Desired Row And Coloumn Location on LCD

#### **2.4.3.8 SIGNAL (SIG\_USART0\_RECV)**

Usart receiver ISR

- < making copy of data from UDR0 in data variable
- < To Regulate the speed
- < To transmit ID to Laptop
- < To transmit speed int value
- < To transmit speed value after decimal point
- < IF third black patch comes then transmit 1 otherwise 0.
- < Image capture process over. Vehicle can go ahead now.

#### **2.4.3.9 void timer1\_init ()**

- < STOP
- < Timer register higher 8 bit to zero
- < Timer register lower 8 bit to zero

- < Output compare register higher 8 bits...not used in this case
- < Output compare register lower 8 bits.....not used in this case
- < Output compare register higher 8 bits...not used in this case
- < Output compare register lower 8 bits.....not used in this case
- < Output compare register higher 8 bits...not used in this case
- < Output compare register lower 8 bits.....not used in this case
- < Using channel A only, but not using output compare mode. WGM: normal mode COM10:1=00
- < No FOC
- < Timer overflow interrupt enabled
- < Setting I flag of status register to one to globally enable all interrupts

#### 2.4.3.10 void timer5\_init ()

- < Stop
- < Counter higher 8-bit value to which OCR5xH value is compared with
- < Counter lower 8-bit value to which OCR5xH value is compared with
- < Output compare register high value for Left Motor
- < Output compare register low value for Left Motor
- < Output compare register high value for Right Motor
- < Output compare register low value for Right Motor
- < Output compare register high value for Motor C1
- < Output compare register low value for Motor C1
- < {COM5A1=1, COM5A0=0; COM5B1=1, COM5B0=0; COM5C1=1 COM5C0=0} For Overriding normal port functionalit to OCRnA outputs. {WGM51=0, WGM50=1} Along With WGM52 in TCCR5B for Selecting FAST PWM 8-bit Mode
- < WGM12=1; CS12=0, CS11=1, CS10=1 (Prescaler=64)

#### 2.4.3.11 void uart0\_init (void)

Function To Initialize UART0 desired baud rate:9600 actual baud rate:9600 (0.0%) char size: 8 bit parity: Disabled

- < disable while setting baud rate
- < set baud rate lo
- < set baud rate hi

#### 2.4.3.12 void velocity (unsigned char *left\_motor*, unsigned char *right\_motor*)

Function for velocity control.

## 2.4.4 Variable Documentation

### 2.4.4.1 unsigned char ADC\_Value

Global Variables

# Index

- ADC\_Conversion
  - AETC\_Robot.c, [13](#)
- adc\_init
  - AETC\_Robot.c, [13](#)
- ADC\_Value
  - AETC\_Robot.c, [16](#)
- AETC\_GSMF.c
  - del\_2s, [4](#)
  - port\_close, [4](#)
  - port\_config, [4](#)
  - port\_open, [4](#)
  - write\_data, [4](#)
- AETC\_GSMS.c
  - del\_2s, [7](#)
  - main, [7](#)
  - port\_close, [7](#)
  - port\_config, [7](#)
  - port\_open, [7](#)
  - write\_data, [7](#)
- AETC\_lcd.c
  - lcd\_cursor, [9](#)
  - lcd\_init, [9](#)
  - lcd\_print, [9](#)
  - lcd\_set\_4bit, [9](#)
  - lcd\_string, [10](#)
  - lcd\_wr\_char, [10](#)
  - lcd\_wr\_command, [10](#)
- AETC\_Robot.c
  - ADC\_Conversion, [13](#)
  - adc\_init, [13](#)
  - ADC\_Value, [16](#)
  - init\_devices, [13](#)
  - main, [13](#)
  - motion\_set, [14](#)
  - port\_init, [14](#)
  - print\_sensor, [14](#)
  - SIGNAL, [14](#)
  - timer1\_init, [14](#)
  - timer5\_init, [15](#)
  - uart0\_init, [15](#)
  - velocity, [15](#)
- del\_2s
  - AETC\_GSMF.c, [4](#)
  - AETC\_GSMS.c, [7](#)
- init\_devices
  - AETC\_Robot.c, [13](#)
- lcd\_cursor
  - AETC\_lcd.c, [9](#)
- lcd\_init
  - AETC\_lcd.c, [9](#)
- lcd\_print
  - AETC\_lcd.c, [9](#)
- lcd\_set\_4bit
  - AETC\_lcd.c, [9](#)
- lcd\_string
  - AETC\_lcd.c, [10](#)
- lcd\_wr\_char
  - AETC\_lcd.c, [10](#)
- lcd\_wr\_command
  - AETC\_lcd.c, [10](#)
- main
  - AETC\_GSMS.c, [7](#)
  - AETC\_Robot.c, [13](#)
- motion\_set
  - AETC\_Robot.c, [14](#)
- port\_close
  - AETC\_GSMF.c, [4](#)
  - AETC\_GSMS.c, [7](#)
- port\_config
  - AETC\_GSMF.c, [4](#)
  - AETC\_GSMS.c, [7](#)
- port\_init
  - AETC\_Robot.c, [14](#)
- port\_open
  - AETC\_GSMF.c, [4](#)
  - AETC\_GSMS.c, [7](#)
- print\_sensor
  - AETC\_Robot.c, [14](#)
- SIGNAL
  - AETC\_Robot.c, [14](#)
- src/AETC\_GSMF.c, [3](#)
- src/AETC\_GSMS.c, [6](#)
- src/AETC\_lcd.c, [8](#)
- src/AETC\_Robot.c, [11](#)
- timer1\_init

AETC\_Robot.c, [14](#)  
timer5\_init  
    AETC\_Robot.c, [15](#)  
  
uart0\_init  
    AETC\_Robot.c, [15](#)  
  
velocity  
    AETC\_Robot.c, [15](#)  
  
write\_data  
    AETC\_GSMF.c, [4](#)  
    AETC\_GSMS.c, [7](#)