

Bluetooth Design document

October 22
2015

This document describes the embedded software design of the Bluetooth module HC-05 Zs-040.

Version 1.0

Revision History

Date	Version	Description	Author	Role
22/10/2015	1.0	Document creation	Ibrahim Mostafa	ES Engineer

Table of Contents

1	Introduction	5
1.1	Purpose	5
1.2	Definitions, Acronyms, and Abbreviations.....	5
1.3	References.....	5
1.4	Overview	5
1.5	Folders and files structure	5
1.6	Connection	5
2	Detailed Design	5
2.1	Initialization Module:	6
2.2	Connection	7
2.3	Error Code	9
3	Driver Functions	10
3.1	Internal Functions	10
3.1.1	EF_BOOLEAN_Bluetooth_GetCharArray.....	10
3.1.2	EF_BOOLEAN_Bluetooth_ReadUntilExpectedKeyword	10
3.1.3	itoa_Convert.....	10
3.2	Global Functions.....	11
3.2.1	EF_BOOLEAN_Bluetooth_ModuleInit.....	11
3.2.2	EF_BOOLEAN_Bluetooth_MasterScan.....	11
3.2.3	EF_BOOLEAN_Bluetooth_MasterConnect.....	11
3.2.4	EF_BOOLEAN_Bluetooth_DisConnect	12
3.2.5	EF_BOOLEAN_Bluetooth_GetModuleStatus	12
3.2.6	EF_BOOLEAN_Bluetooth_SendData	12
3.2.7	EF_BOOLEAN_Bluetooth_GetDataIfExist.....	13

Table of Figures

1 Introduction

1.1 Purpose

The purpose of this document is to describe the detailed design of the Bluetooth HC-05 ZS-040 module and how it works.

1.2 Definitions, Acronyms, and Abbreviations

ACK

Acknowledgement

1.3 References

Item	Name	link
[1]	Embedded_Bluetooth_Serial_Communication_Module	
[2]	Good tutorial	http://blog.zakkemble.co.uk/getting-bluetooth-modules-talking-to-each-other/

1.4 Overview

Bluetooth serial module is used for converting serial port to Bluetooth. These modules have two modes: master and slaver device.

1.5 Folders and files structure

Bluetooth module was implemented by three files: EF_Bluetooth.c, EF_Bluetooth_cfg.h and EF_Bluetooth.h .

1.6 Connection

The Bluetooth requires 5 volts for power.

The Bluetooth needs to communicate via serial at 3.3V and does not have 5V tolerant inputs, so you need level conversion to communicate with a 5V microcontroller like most Arduinos use.

FTDI cable or USB to TTL converter this module give communication from TTL to USB.

BlueTerm or SENA Bterm Android software, get it from google play on your android phone

NOTE: the module need around 30 mA when pairing and 8mA when non-pairing to receive AT Command.

2 Detailed Design

The module works with AT commands via UART protocol.

2.1 Initialization Module:

Using Some Commands to define the Mode as you can make master or slave, define its name, define password, get Module Version and get MAC Address of Module.

1. Test UART connection

COMMAND	RESPONSE
AT↵	OK↵

2. Reset Device

COMMAND	RESPONSE
AT+RESET↵	OK↵

3. Query firmware version

COMMAND	RESPONSE
AT+VERSION?↵	+VERSION:<VER>↵ OK↵

where <VER> = Version Number

4. Restore settings to Factory Defaults

COMMAND	RESPONSE
AT+ORGL↵	OK↵

Restore to the following settings:

Device Class: 0

Inquiry Code: 0x009e8b33

Device Mode: Slave

Binding Mode: SPP

UART: 38400bps, 8 bit, 1 stop bit, no parity

Pairing Code: 1234

Device Name: H-C-2010-06-01

COMMAND	RESPONSE
AT+ROLE?↵	+ROLE:<role>↵ OK↵
AT+ROLE=<role>↵	OK↵

where <role>
0 - Slave (default)
1 - Master
2 - Slave-Loop

Slave - EGBT-045MS acts as discoverable wireless UART device ready for transparent data exchange.

Master - Scans for a remote bluetooth (slave) device, pairs, and setup connection for a transparent data exchange between devices

Slave-Loop - Data loop-back Rx-Tx. Used mainly for testing.

Example: Set EGBT-045MS in master role

Bluetooth address in NA:UAP:LAP format = 0002:72:0A3C7F

From Host controller:

AT+ROLE=1↵

EGBT-045MS response

+ROLE:1↵

OK↵

2.2 Connection

The default situation of HC-04 is slave mode. If you need master mode, please state it clearly or place an order for HC-04-M directly. The naming rule of HC-06 is same.

First you have to connect your FTDI module (or USB-TTL converter) directly to your laptop, and make these wiring to between your FTDI and BT module:

FTDI BT

RX – TX

TX – RX

VCC – VCC

GND –GND

Now you should see red light on BT module flashing rapidly and happy, now search BT module in your android phone, you can see your Bluetooth device name, this is the BT module.

Just click and connect it, if you type 0000 or 1234, should be 1234, you should pair it successfully. And the red light

flash slowly!

Now turn on Terminal software on laptop, use default setting (9600 baud if not pressing switch when powered else 38400, parity = none, stop bit = 1), select the right com and connect, then turn on SENA Bterm on phone, go Bluetooth management – connect to your Bluetooth module, now you can type any words on either SENA Bterm or Terminal to see the communication.

HC-05 master device has no memory before the first use. If the password is correct, the mater device will make pair with the slave device automatically in the first use. In the following use, the master device will remember the Bluetooth address of the last paired device and search it.

Pin 34:

Mode switch input. If it is input low level, the module is at paired or communication mode. If it's input high level, the module will enter to AT mode. Even though the module is at communication, the module can enter to the AT mode if PIN34 is input high level. Then it will go back to the communication mode if PIN34 is input low level again.

SO , Pin 34 always should be high 3.3v except during Communication Mode to enable data transfer.

27	AT+STATE	Query Current Status of the Device
28	AT+INIT	Initialize SPP Profile
29	AT+INQ	Query Nearby Discoverable Devices
30	AT+INQC	Cancel Search for Discoverable Devices
31	AT+PAIR	Device Pairing
32	AT+LINK	Connect to a Remote Device
33	AT+DISC	Disconnect from a Remote Device

27. Query Current Status of the Device

COMMAND	RESPONSE
AT+STATE?↵	+STATE:<stat>↵ OK↵

where

<stat> = Current Status, any one of the following:

INITIALIZED
READY
PAIRABLE
PAIRED
INQUIRING
CONNECTING
CONNECTED
DISCONNECTED
UNKNOWN

2.3 Error Code

The Module may return error Code.

ERROR CODE	VERBOSE
0	Command Error/Invalid Command
1	Results in default value
2	PSKEY write error
3	Device name is too long (>32 characters)
4	No device name specified (0 lenght)
5	Bluetooth address NAP is too long
6	Bluetooth address UAP is too long
7	Bluetooth address LAP is too long
8	PIO map not specified (0 lenght)
9	Invalid PIO port Number entered
A	Device Class not specified (0 lenght)
B	Device Class too long
C	Inquire Access Code not Specified (0 lenght)
D	Inquire Access Code too long
E	Invalid Iquire Access Code entered
F	Pairing Password not specified (0 lenght)
10	Pairing Password too long (> 16 characters)
11	Invalid Role entered
12	Invalid Baud Rate entered
13	Invalid Stop Bit entered
14	Invalid Parity Bit entered
15	No device in the Pairing List
16	SPP not initialized
17	SPP already initialized
18	Invalid Inquiry Mode
19	Inquiry Timeout occurred
1A	Invalid/zero lenght address entered
1B	Invalid Security Mode entered
1C	Invalid Encryption Mode entered

3 Driver Functions

3.1 Internal Functions

3.1.1 EF_BOOLEAN_Bluetooth_GetCharArray

<i>Format</i>	EF_BOOLEAN_Bluetooth_GetCharArray(U8_t UART_Number ,U8_t * ReturnedArray, U16_t NewLine_Numbers)
<i>Description</i>	This function used to receive from UART array until defined given NewLine Number or until number of defined tries or until reach to the Max Size of RX Buffer Array.
<i>Argument</i>	UART_Number : Number ,max. digits of it should be equaled MAX_DIGIT_FOR_ITOA ReturnedArray : pointer to the returned Array NewLine_Numbers : Max New Line receive character to stop when reaching it.
<i>Return value</i>	BOOLEAN to check for Errors

3.1.2 EF_BOOLEAN_Bluetooth_ReadUntilExpectedKeyword

<i>Format</i>	EF_BOOLEAN_Bluetooth_ReadUntilExpectedKeyword (U8_t* ExpectedKeyword, U8_t ExpectedKeyword_Length, U16_t TimeOut)
<i>Description</i>	EF_BOOLEAN_ESP8266_ReadUntilExpectedKeyword This function used to receive Array by UART, and stop until Expected keyword received or when timeout occurred
<i>Argument</i>	ExpectedKeyword : Expected array to stop when find it or when time out ExpectedKeyword_Length : length of ExpectedKeyword Array. TimeOut : mili seconds (Ex: if timeout 5000 means 5sec), using to extract from this function if ExpectedArray was not found
<i>Return value</i>	TRUE: means OK, found this Expected word in Receiving progress. FALSE : means time is out and not found the ExpectedArray

3.1.3 itoa_Convert

<i>Format</i>	itoa_Convert (U32_t Number , U8_t * NumberASCII_ptr ,U8_t* NumberOfDigits_ptr)
<i>Description</i>	This function used to convert integer number to ASCII
<i>Argument</i>	U16_t Number : integer Number

	U8_t * NumberASCII_ptr : pointer to the ASCII number after Conversion NumberOFDigits_ptr: pointer to size NumberASCII_ptr array
<i>Return value</i>	None.

3.2 Global Functions

3.2.1 EF_BOOLEAN_Bluetooth_ModuleInit

<i>Format</i>	EF_BOOLEAN_Bluetooth_ModuleInit (BOOLEAN bMaster, U8_t* Password ,U8_t* ModuleName , U8_t ModuleName_Length);
<i>Description</i>	This function used to initialise UART, Enable AT Command Pin (make it 3.3v to at command and 0 for Communication Mode) , Timer and Send some Init Commands
<i>Argument</i>	bMaster : 1 to make Bluetooth Module to master and 0 to be slave Password : Set Bluetooth Module Password (4 Number characters (ASCII) like: "1234") ModuleName: Set Bluetooth Module Name
<i>Return value</i>	BOOLEAN to check for Errors

3.2.2 EF_BOOLEAN_Bluetooth_MasterScan

<i>Format</i>	EF_BOOLEAN_Bluetooth_MasterScan (U8_t ScanDevicesNumber , U8_t TimeOut, U8_t* ReturnedAddresses);
<i>Description</i>	This function used to query/search for the nearby discoverable devices, print and return the MAC Add. which is found.
<i>Argument</i>	ScanDevicesNumber: max number of devices to stop searching when reaching it. TimeOut :max time to stop searching when reaching it. ReturnedAddresses: MAC Add. of existing devices, (you should give it pointer to Large array Like 50 character)
<i>Return value</i>	BOOLEAN to check for Errors

3.2.3 EF_BOOLEAN_Bluetooth_MasterConnect

<i>Format</i>	EF_BOOLEAN_Bluetooth_MasterConnect (U8_t SlaveMAC_Add , U8_t TimeOut, U8_t* ReturnSlaveName);
<i>Description</i>	This function used to pair , connect to defined slave device with it's MAC ADD , print and return Slave Device name.

<i>Argument</i>	SlaveMAC_Add : 14 characters (12 of them is the MAC ADD of the wanted Slave device. give it as : 4char,2char,6char == "1CAF,05,D69CE9") TimeOut : max time to enable Slave to enter Password and pairing ReturnSlaveName: return Slave Device name.
<i>Return value</i>	BOOLEAN to check for Errors

3.2.4 EF_BOOLEAN_Bluetooth_DisConnect

<i>Format</i>	EF_BOOLEAN_Bluetooth_DisConnect ();
<i>Description</i>	This function used to disconnect Connection whether Module was Master or Slave
<i>Argument</i>	None.
<i>Return value</i>	BOOLEAN to check for Errors

3.2.5 EF_BOOLEAN_Bluetooth_GetModuleStatus

<i>Format</i>	EF_BOOLEAN_Bluetooth_GetModuleStatus (U8_t* Status);
<i>Description</i>	This function used to print and return the Module Status returned +STATE:<stat>\r\nOK\r\n , <stat> are 9 status
<i>Argument</i>	Status: pointer to the returned status.
<i>Return value</i>	return 1 if INITIALIZED : Slave or Master after power up return 2 if READY return 3 if PAIRABLE : Slave after AT+INIT return 4 if PAIRED : Master after pairing or binding return 5 if INQUIRING : Master after quering return 6 if CONNECTING : Master after pairing when Communication Enable (0 v to pin) return 7 if CONNECTED : Slave or Master after linking return 8 if DISCONNECTED: Master or Slave return 9 if UNKNOWN return 0 if error

3.2.6 EF_BOOLEAN_Bluetooth_SendData

<i>Format</i>	EF_BOOLEAN_Bluetooth_SendData (U8_t* Data, U8_t DataLength);
<i>Description</i>	This function used to send data
<i>Argument</i>	Data : pointer to send data number of characters
<i>Return value</i>	return FALSE if the status is not connected return TRUE if ok

3.2.7 EF_BOOLEAN_Bluetooth_GetDataIfExist

<i>Format</i>	EF_BOOLEAN_Bluetooth_GetDataIfExist (U8_t* RxData_ptr);
<i>Description</i>	This function used to receive data if exist
<i>Argument</i>	RxData_ptr: pointer to returned data.
<i>Return value</i>	return FALSE if the status is not connected, return TRUE if ok