XLR8 Bluetooth Receiver ModuleDocument

Electronics Club

IIT Bombay

This page is intentionally left blank

About

This document describes the implementation of Bluetooth controlled port using Attiny2313A.

This document (program) is limited to 4 bit output , but can be easily changed upto 8 bits & beyond by configuring the data packet protocol (user decided).

Index

Contents

Introduction

Bluetooth controlled port is the digital output access over Bluetooth (in this case 4 bit access) and is achieved using a Bluetooth module (HC05/06) and a microcontroller (ATtiny2313A).

This can be controlled through any Bluetooth device which can access the connection to the HC05/06.

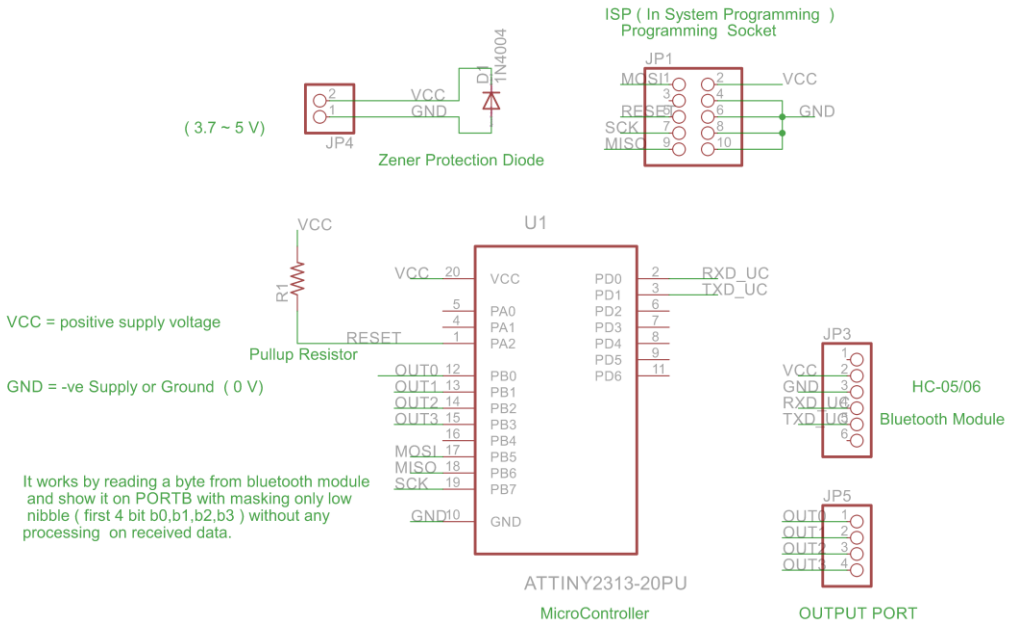
Hardware

Following hardware is used

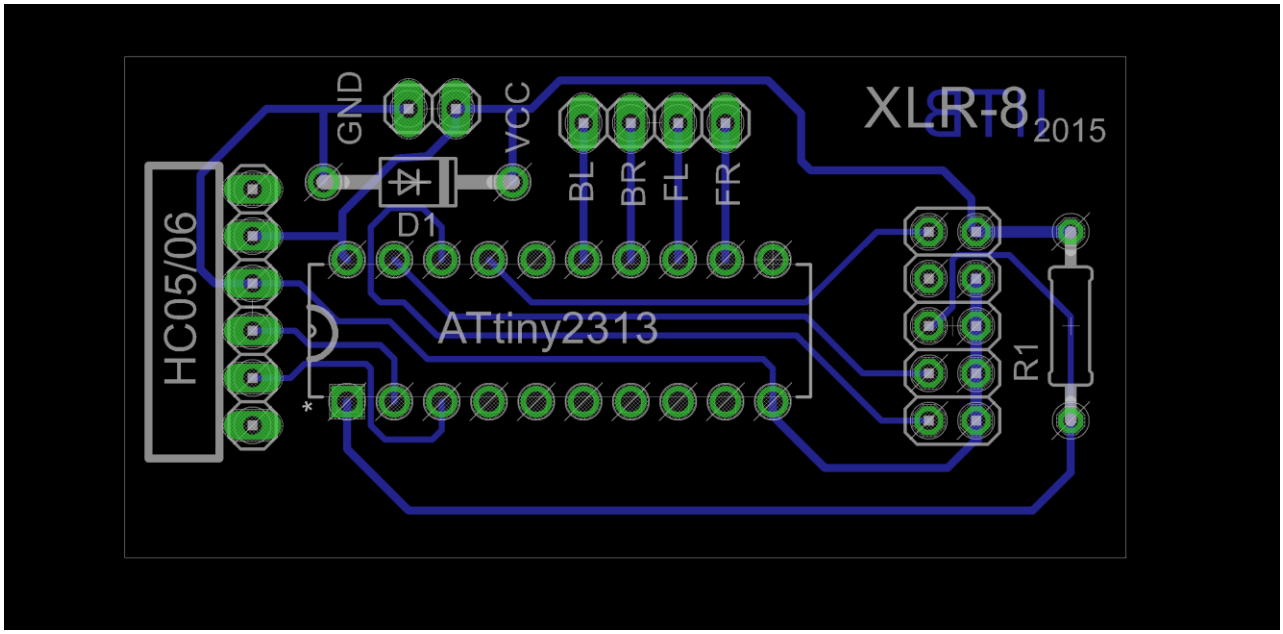
- HC05/06 Bluetooth Module
- Attiny 2313A microcontrollerd

Schematic & PCB Diagram

Schematic & PCB designing is done in EAGLE7.2.0 Light Version



Schematic diagram



PCB top-bottom-silkscreen

Eagle files can be found at
https://github.com/ajinkyagorad/XLR8_Receiver/tree/master/EAGLE_Files

Software

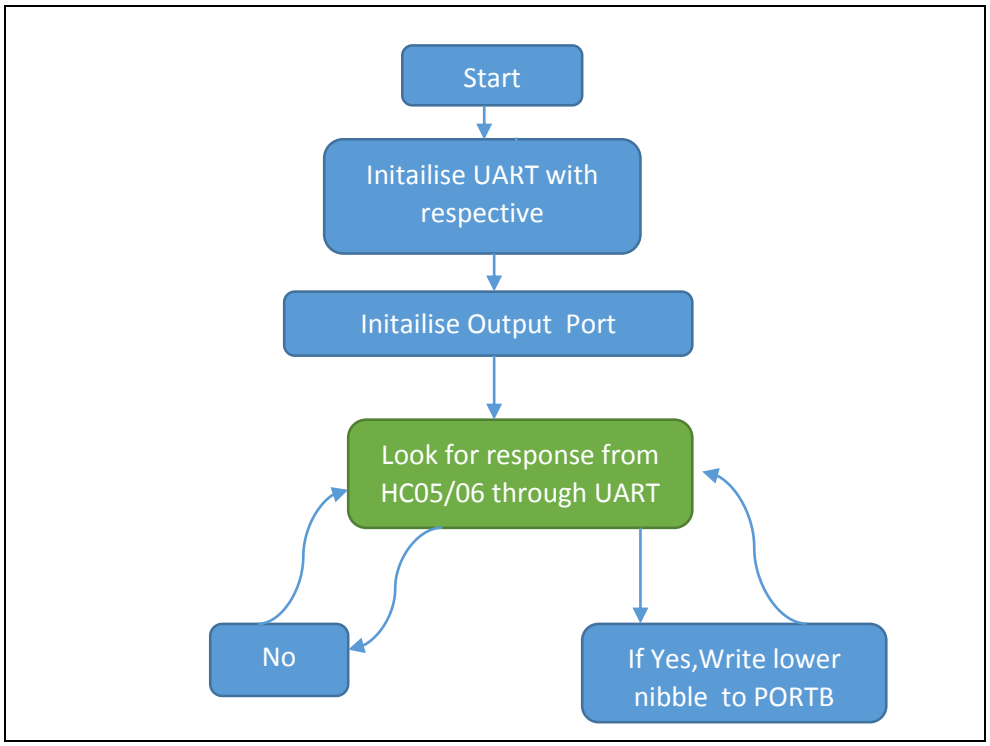
Software (program) for ATtiny2313 is written in Atmel Studio 6.0

```
/*
 * XLR8_module.c
 *
 * Created: 20-07-2015 10:07:35 PM
 *  Author: Ajinkya & Meet
 */

#include <avr/io.h>                        // definations for registers
//default frequency of operation 1M
void USARTInit(unsigned int ubrr_value, uint8_t x2, uint8_t stopbits) {
       //from datasheet :
       // Set baud rate
       UBRRL = ubrr_value & 255;
       UBRRH = ubrr_value >> 8;
       // Frame Format: asynchronous, 8 data bits, no parity, 1/2 stop bits
       UCSRC = _BV(UCSZ1) | _BV(UCSZ0);
       if(stopbits == 2) UCSRC |= _BV(USBS);
       if(x2) UCSRA = _BV(U2X); // 2x
       // USART Data Register Empty Interrupt Enable
       UCSRB = _BV(UDRIE);
       // Enable The receiver and transmitter
       UCSRB |= _BV(RXEN) | _BV(TXEN);
}

int main() {
```

```
        USARTInit(5, 0, 1); // initialise uart to baud rate 9600, no parity , 1
stopbit
        DDRB=0xFF;                  // make all pins on port B as output
        PORTB  = 0x00;              // initially make all pins low
        while(1)                    //endless loop
                while( !(UCSRA & (1<<RXC)) );      // wait for data to be received in
uart UDR register
                char temp = UDR;                            // copy data,
                PORTB  = UDR&(0X0F);                              // write data directly
to port
        }
}
```

For more information related to microcontroller refer http://www.atmel.com/images/doc2543.pdf

Program Flowchart



Simplified Flow chart for Receiver

This program directly writes the data received through Bluetooth Module (1 byte) to PORTB of the ATtiny 2313A.

The schematic diagram (circuit diagram) given is for low nibble (4 lower bits) of PORTB.

The default clock frequency of 1MHz (internal RC oscillator) is used as a clock source.

Programming

Avrdude programming command using usbasp is

```
avrdude –c usbasp –p attiny2313 –U flash:w:XLR8_module.hex
```

For more details refer 'XLR8_Bluetooth_Programming_Guide.pdf' at
https://github.com/ajinkyagorad/XLR8_Receiver/tree/master/Document

PCB Soldering Tutorial

Refer 'XLR8_Bluetooth_RF_module_Soldering_guide.pdf' for picture guide at
https://github.com/ajinkyagorad/XLR8_Receiver/tree/master/Document

Future Work

- This can be improved for larger number of bit access by defining a Bluetooth UART packet protocol, or up to n bits by masking for first n bits (n < 8) according to application.
- If possible to change the firmware on the BC417 chip on Bluetooth module for directly using HC05 pins remotely without the need for the external microcontroller (see http://byron76.blogspot.in/). Which can enable many functions like sampling voltage remotely without uC and at a low cost.

Reference Links

- Github Project Link https://github.com/ajinkyagorad/XLR8_Receiver
- Eagle files https://github.com/ajinkyagorad/XLR8_Receiver/tree/master/EAGLE_Files
- Datasheet for ATtiny2313A http://www.atmel.com/images/doc2543.pdf
- Guides documents
  https://github.com/ajinkyagorad/XLR8_Receiver/tree/master/Document
- Interesting future work http://byron76.blogspot.in/