Patrick Harrington
Maxim Allen
ECEN 2020
Homework 6
October 21, 2013

1 Serial Communication

(a) Echo Program

```
/*****************************************************************
* Take a character when it is typed in, and repeats it back
* —Does not use Serial Library
* —Uses UCSR0C register for async communication
**A**************************************************************/
unsigned char c = 0;


void setup(){
  USART_Init();
}

/*****************************************************************
*  Code for setting up USART on ATmega328
*****************************************************************/

void USART_Init(void){
  // CLEAR THE REGISTERS
  // The absence of this part crippled our code and poured out
  // copious amounts of spaghetti on our Serial Monitors.
  // It was a dark time, I tell you...
  UCSR0A = 0;
  UCSR0B = 0;
  UCSR0C = 0;
  //  Set baud rate for 9600, Clock speed is 16MHz
  UBRR0H = 0;
  UBRR0L = 207; // can also be written as 0xCF

  //  Set U2X to double the transmission speed
  UCSR0A |= (1<<U2X0);
  //  Enable receiver and transmitter
  UCSR0B |= (1<<TXEN0)|(1<<RXEN0);
  //  Set frame format: asyncronous, no parity, 1 stop bit, 8bit data
  UCSR0C |= (1<<UCSZ01)|(1<<UCSZ00);
}

/*****************************************************************
*  Code for setting up USART Receive on ATmega328
```

```
***************************************************************/

unsigned char USART_Receive(void){
  while(!(UCSR0A & (1<<RXC0)) );
  //  Get and return received data from the buffer
  return UDR0;
}

/***************************************************************
*  Code for setting up USART Transmit on ATmega328
***************************************************************/

void USART_Transmit( unsigned char datum ){
  while ( !(UCSR0A & (1<<UDRE0)) );
  //  place data in buffer
  UDR0 = datum;
}

void loop(){

  c = USART_Receive();
  //  this code grabs a character
  //  this code sends it to the serial monitor
  USART_Transmit(c);
}
```

(b) Changing Baud rate to 115200

By manipulating `UBRR0H` and `UBRR0L`, the Baud rate can be set to 115200. Specifically, this can be done by setting `UBRR0H` to 0 and `UBRR0L` to 16 as listed in the Data sheet. (Note that in your favorite serial monitor, the Baud rate must be adjusted, too).

2 Playing Music

Yes; I got this working.

3 Communicating with the MP3 Board

```
#include <SPI.h>

// set pin 10 as the slave select
#define CS 6
#define DREQ 2

void setup() {
  pinMode (CS, OUTPUT);
  pinMode (DREQ, INPUT);
  Serial.begin(57600);
  SPI.begin();
  Serial.println("ENTER␣AN␣ADDRESS");
}

void loop() {
  unsigned int recChar;  //Recieve a char
  if (Serial.available())
  {
    recChar = Serial.parseInt();  //Wait for SPI data
    unsigned int readReg = readRegister((unsigned char)recChar);
    Serial.println(readReg, HEX);
  }
}
unsigned int readRegister(unsigned char address)
{
  while(!digitalRead(DREQ));
  digitalWrite(CS,LOW);
  SPI.transfer(0x03); //Read instruction
  SPI.transfer(address);  //Read a register, this input is buffered

  char response1 = SPI.transfer(0xFF); //Read byte, send full byte, MSB
  while(!digitalRead(DREQ));
  char response2 = SPI.transfer(0xFF); //Read byte, send full byte, LSB
  while(!digitalRead(DREQ));
  digitalWrite(CS,HIGH);


  int resultvalue = response1 << 8;
  resultvalue |= response2;
  return resultvalue;
}
```