//SPI Master

#include <SPI.h>

void setup (void) {

Serial.begin(115200); //set baud rate to 115200 for usart

digitalWrite(SS, HIGH); // disable Slave Select

SPI.begin ();

SPI.setClockDivider(SPI\_CLOCK\_DIV8);//divide the clock by 8

}

void loop (void) {

char c;

digitalWrite(SS, LOW); // enable Slave Select

// send test string

for (const char \* p = "Hello, world!\r" ; c = \*p; p++)

{

SPI.transfer (c);

Serial.print(c);

}

digitalWrite(SS, HIGH); // disable Slave Select

delay(2000);

}

//////////////////////////////////////////////////////////////////////

//SPI Slave

#include <SPI.h>

char buff [50];

volatile byte indx;

volatile boolean process;

void setup (void) {

Serial.begin (115200);

pinMode(MISO, OUTPUT); // have to send on master in so it set as output

SPCR |= \_BV(SPE); // turn on SPI in slave mode

indx = 0; // buffer empty

process = false;

SPI.attachInterrupt(); // turn on interrupt

}

ISR (SPI\_STC\_vect) // SPI interrupt routine

{

byte c = SPDR; // read byte from SPI Data Register

if (indx < sizeof buff) {

buff [indx++] = c; // save data in the next index in the array buff

if (c == '\r') //check for the end of the word

process = true;

}

}

void loop (void) {

if (process) {

process = false; //reset the process

Serial.println (buff); //print the array on serial monitor

indx= 0; //reset button to zero

}

}

