

```

/*
 * Program to transfer/receive serial data to/from 1802
 */
#include <avr/io.h>
#include <avr/portpins.h>

const byte Interrupt1802 = A2;
const byte ReadByteFrom1802 = A3;
const byte Initiate_DMA_IN = A4;
const byte LatchByte = A5;
const byte ByteFrom1802 = 2;
const byte ACKFrom1802 = 3;
const byte LOAD_State_Pin = 12;
volatile byte ByteFrom1802Flag = 0;
volatile byte ACKfrom1802Flag = 0;
volatile byte TXDataReg;
byte RXDataReg;
byte LOAD_State;

void InterruptFrom1802() {
  //Read Byte from 1802 OUTPUT 74373
  digitalWrite(ReadByteFrom1802,LOW);
  PORTC&=0xF7;
  TXDataReg=(PIND & 0xF0) | (PINB & 0x0F);
  PORTC|=0x08;
  ByteFrom1802Flag=1;
}

void ACKInterrupt(){
  ACKfrom1802Flag=1;
}

void setup() {
  //Config Analog pins as Outputs
  pinMode(Interrupt1802,OUTPUT);
  pinMode(ReadByteFrom1802,OUTPUT);
  pinMode(LatchByte,OUTPUT);
  pinMode(Initiate_DMA_IN,OUTPUT);
  //Interrupt1802 and ReadByteFrom1802 must be HIGH as they are active LOW signals
  //LatchByte and Initiate_DMA_IN is active HIGH signal
  PORTC|=0x0C;
  //Serial.begin(921600);
  Serial.begin(230400);
  Serial.println("ACIA Ready");
  pinMode(ByteFrom1802, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(ByteFrom1802),InterruptFrom1802,RISING);
}

```

```

pinMode(ACKFrom1802, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(ACKFrom1802),ACKInterrupt,RISING);
}

void loop() {
  while (!Serial.available()){
    //While waiting for data from terminal/device, send any new byte in TXDataReg
    if (ByteFrom1802Flag){
      Serial.write(TXDataReg);
      ByteFrom1802Flag=0;
    }
    ProcessLOADState();
  }
  RXDataReg=Serial.read();
  //Send RXDataReg byte to 1802
  LatchByteInto373s();
  ACKfrom1802Flag=0;
  //Initiate 1802 INT
  PORTC&=0xFB; //Clear Interrupt1802 pin
  while (!ACKfrom1802Flag){
    //Wait until 1802 ACKnowledges it has read the byte
  }
  PORTC|=0x04; //Set Interrupt1802 pin
}

void LatchByteInto373s() {
  DDRD|=0xF0; //Pins 4,5,6,7 OUTPUTS
  DDRB|=0x0F; //Pins 8,9,10,11 OUTPUTS
  delayMicroseconds(10);
  PORTD=RXDataReg & 0xF0;
  PORTB=RXDataReg & 0x0F;
  delayMicroseconds(10);
  //Latch byte RXDataReg in INPUT 74373
  PORTC|=0x20; //Set Latchbyte pin
  PORTC&=0xDF; //Clear Latchbyte pin
  DDRD&=0x0F; //Pins 4,5,6,7 INPUTS
  DDRB&=0xF0; //Pins 8,9,10,11 INPUTS
}

void ProcessLOADState() {
  LOAD_State=digitalRead(LOAD_State_Pin);
  if (!LOAD_State) {
    Serial.println("LOAD State");
  }
  while(!LOAD_State) {
    byte OpCode=0;

```

```

byte OpCode2=0;
while (!Serial.available()){
    //Wait for first data byte from terminal/device
    //Exit sub if no longer in LOAD State
    LOAD_State=digitalRead(LOAD_State_Pin);
    if (LOAD_State) {
        break;
    }
}
if (LOAD_State) {
    Serial.println("ACIA Ready");
    break;
}
RXDataReg=Serial.read();
if (isHexadecimalDigit(RXDataReg)) {
    if (RXDataReg>=0x30 && RXDataReg<=0x39) {
        OpCode+=RXDataReg-0x30;
    }else{
        if (RXDataReg>=0x41 && RXDataReg<=0x46) {
            OpCode+=RXDataReg-0x37;
        }else{
            if (RXDataReg>=0x61 && RXDataReg<=0x66) {
                OpCode+=RXDataReg-0x57;
            }else{
                OpCode=0;
            }
        }
    }
}
}else{
    OpCode=0;
}
OpCode=OpCode*0x10;
while (!Serial.available()){
    //Wait for 2nd data byte
    //Exit sub if no longer in LOAD State
    LOAD_State=digitalRead(LOAD_State_Pin);
    if (LOAD_State) {
        break;
    }
}
if (LOAD_State) {
    Serial.println("ACIA Ready");
    break;
}
RXDataReg=Serial.read();
if (isHexadecimalDigit(RXDataReg)) {

```

```

if (RXDataReg>=0x30 && RXDataReg<=0x39) {
  OpCode2+=RXDataReg-0x30;
}else{
  if (RXDataReg>=0x41 && RXDataReg<=0x46) {
    OpCode2+=RXDataReg-0x37;
  }else{
    if (RXDataReg>=0x61 && RXDataReg<=0x66) {
      OpCode2+=RXDataReg-0x57;
    }else{
      OpCode2=0;
    }
  }
}
}else{
  OpCode2=0;
}
RXDataReg=OpCode+OpCode2;
LatchByteInto373s();
//Initiate 1802 DMA_IN
PORTC&=0xEF;
digitalWrite(Initiate_DMA_IN,LOW);
PORTC|=0x10; //Set Initiate_DMA_IN pin
LOAD_State=digitalRead(LOAD_State_Pin);
if (!LOAD_State) {
  Serial.println(">");
}else{
  Serial.println("ACIA Ready");
}
}
}

```