CPE301 – SPRING 2022

MIDTERM 1

Student Name: Angelo Nolasco
Student #: 5005497011
Student Email: Nolasco@unlv.nevada.edu
Primary Github address: angelon3121@gmail.com
Directory:

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmel Studio 7.0

- Simulator
- Debugger

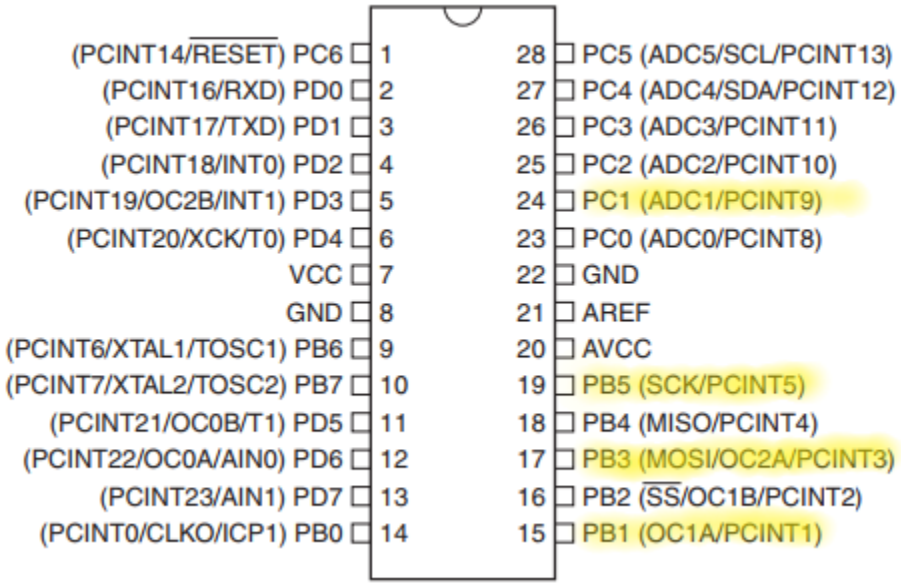
Atmega328PB-Xmini

Multi-Function Shield

- LEDS
- Switches

ATMEGA328

Port Pin Used



2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

```
/************Global Variables*******************/
// The inputted commands are never going to be
// more than 8 chars long. Volatile for the ISR.
volatile unsigned char data_in[8];
volatile unsigned char command_in[8];
```

```
volatile unsigned char data_count;
volatile unsigned char command_ready;

// Variables to hold current settings
unsigned int sensitivity = 223;
/*********************************************/
/************Function Protype*****************/
void USART0_Put(uint8_t data);
void USART0_PutString(char *ptr);
void copy_command ();
void timer1(void);
void process_command();
void Fade_LED(void);
/*********************************************/
/************Function Implementation**********/
void USART0_Put(uint8_t data){

//Checking to see if USART TX buffer is empty for new data
while(!(UCSR0A & (1<<UDRE0)));

//Initiating transfer
UDR0 = data;

} // USART0_Put()
void USART0_PutString(char *ptr){

while(*ptr){                    // Loop until end of string (*s = '\0')
USART0_Put(*ptr++);             // Send the character and point to the next one
}

} // USART0_PutString()

void copy_command ()
{
// The USART might interrupt this - don't let that happen!
ATOMIC_BLOCK(ATOMIC_FORCEON) {
// Copy the contents of data_in into command_in
memcpy(command_in, data_in, 8);

// Now clear data_in, the USART can reuse it now
memset(data_in[0], 0, 8);
}
}

void timer1(void){//the function use timer1 in CTC mode
```

```
DDRB |= (1 << DDB3) ; //configure PB3 as output
TCCR1B = (1 << CS12); //set the pre-scalar as 256
OCR1A = 46875; //750ms delay
TCNT1 = 0;
while (1)
{
//If flag is set toggle the led
while ((TIFR1 & (1 << OCF1A)) == 0); // wait till the timer overflow flag is SET
PORTB ^= (1 << DDB3);
TCNT1 = 0;
TIFR1 |= (1 << OCF1A) ; //clear timer1 overflow flag
}

}

void Fade_LED(void){

}


void process_command()
{
switch (command_in[0]) {
case 'h'://print the help menu
USART0_PutString("Help Screen\n");
USART0_PutString("‘o’ - turns ON LED at PB5,‘O’ turns OFF the LED                    at PB5
\n");
USART0_PutString("‘p’ - Blink (on-off) the LED PB3,‘P’ turns off the                    LED
PB3 \n");
USART0_PutString("‘f’ - fade the intensity of LED PB1,‘F’ turns off the                    LED
PB1 \n");
USART0_PutString("‘b’ – reads the status of the switch at PC1 \n");
break;
case 'o': //turn on the PB5 LED
DDRB |= (1<<5);//set PB5 as output
PORTB |=(1<<5);//turn on LED at PB5
break;
case 'O': //turn off the PB5 LED
DDRB &=~(1<<5);//turn off LED PB5
PORTB &=~(1<<5);
break;
case 'p': //make PB3 LED blink
timer1();//call timer1 function
break;
case 'P': //turn off the PB3 LED
DDRB &=~(1<<DDB3);//turn off LED PB3
```

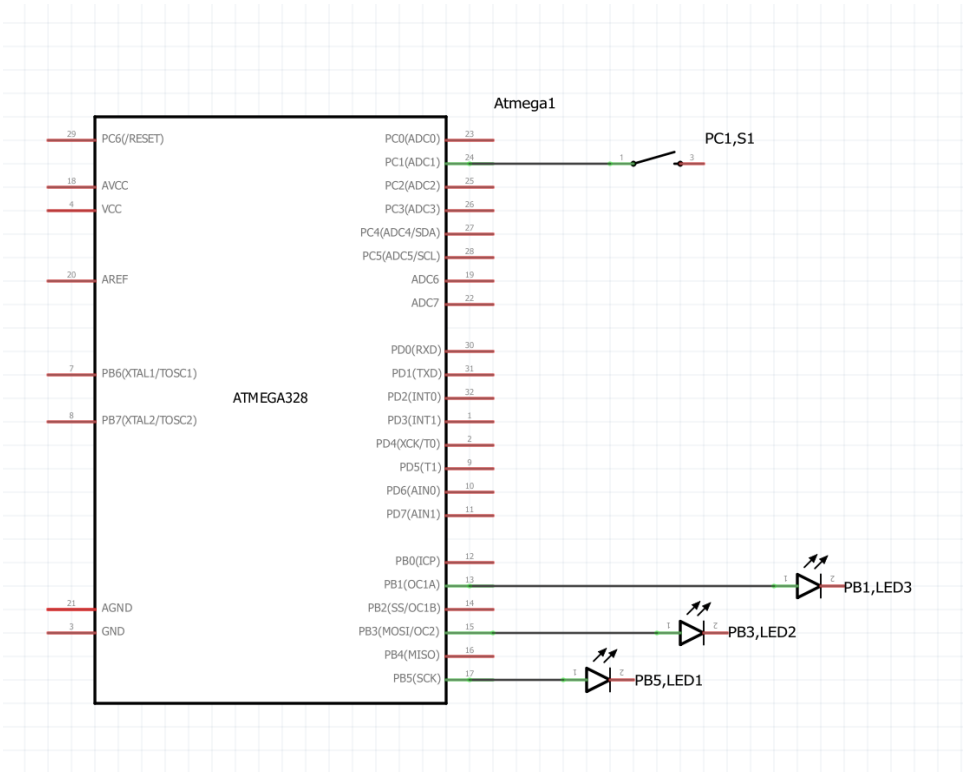
```
PORTB &=~(1<<DDB3);
break;
case 'f': //make PB1 LED fade
Fade_LED();//call fade_led function
break;
case 'F': //turn off PB1 LED
DDRB &=~(1<<1);//turn off LED PB1
PORTB &=~(1<<1);
break;
case'b': //read value from switch PC1

break;

default:
break;
}//end of switch statement
}
```

3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

N/A

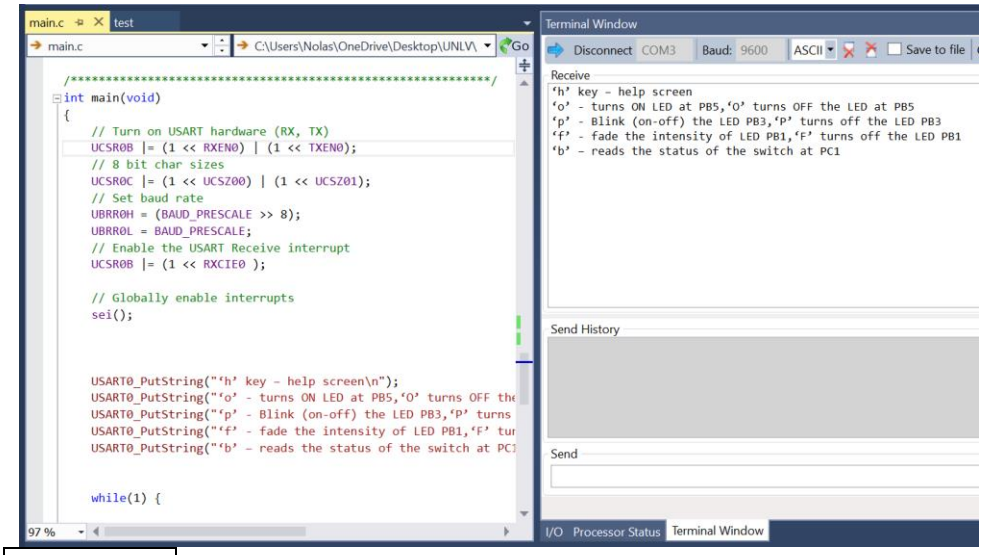
4. SCHEMATICS

Use fritzing.org

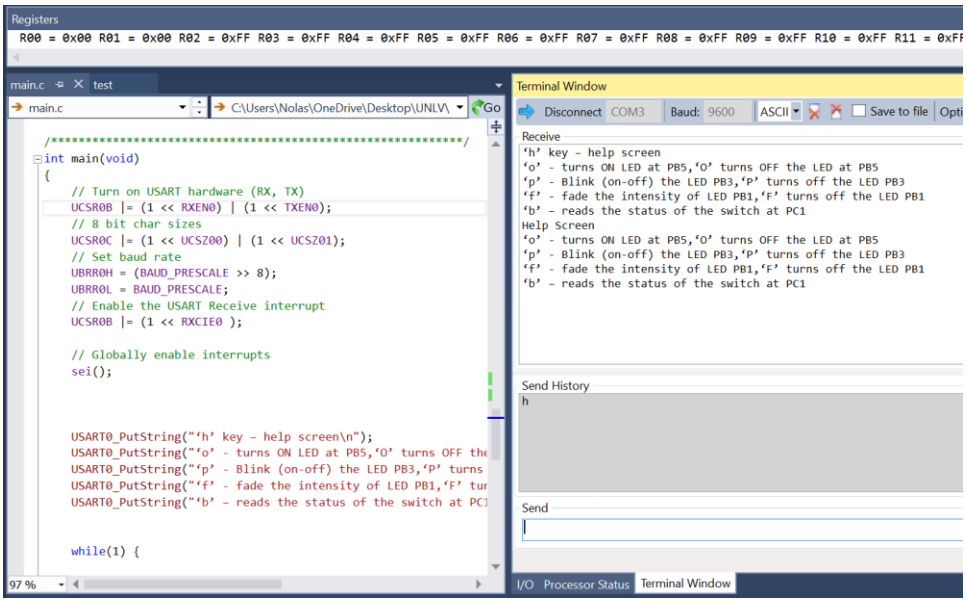


5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

On-reboot



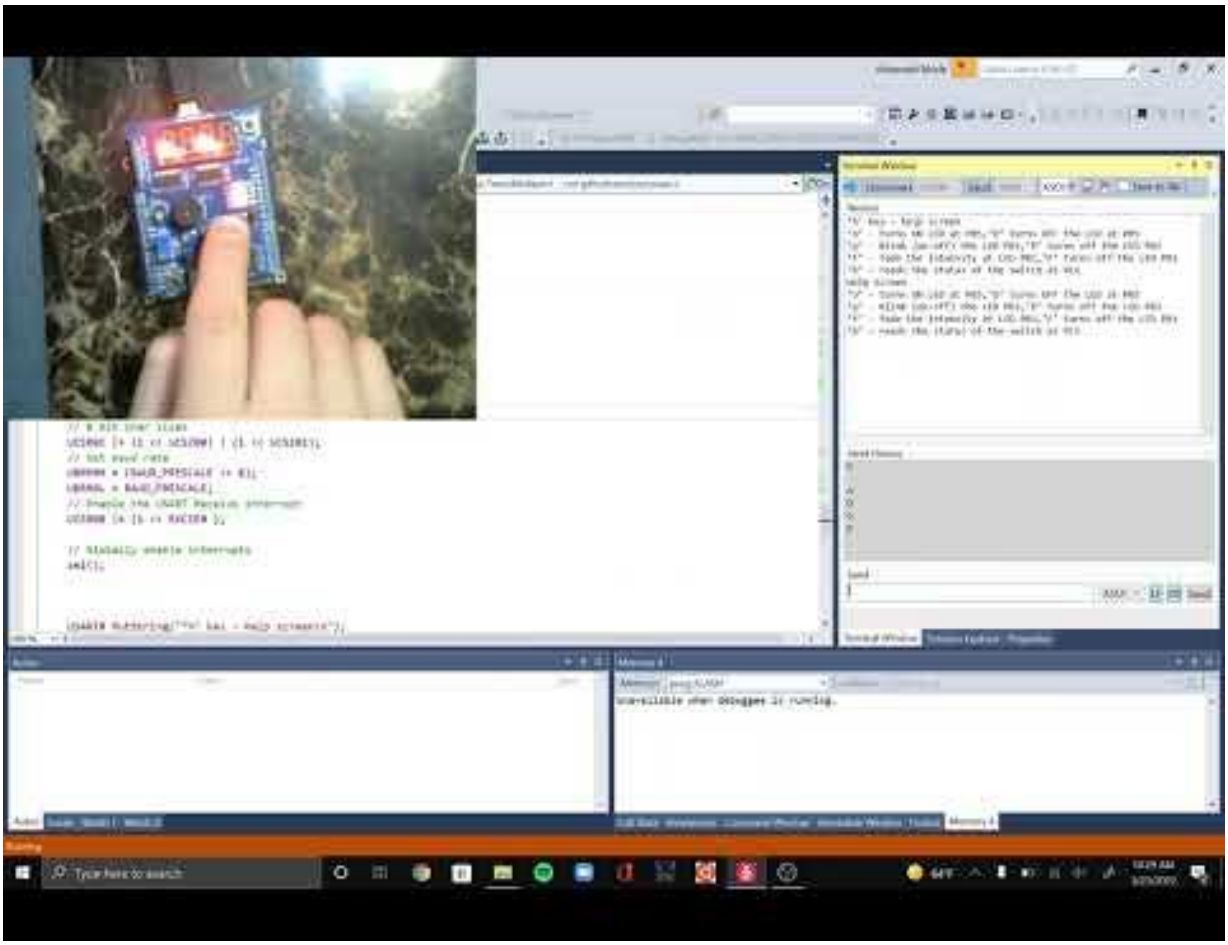
'h'-key



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

7. VIDEO LINKS OF EACH DEMO

[Mid-Term1 C Program]



8. GITHUB LINK OF THIS DA

https://github.com/AngeloNol/DA_submission

"*This assignment submission is my own, original work*".
Angelo Nolasco