Noor blum

Design Assignment DA5

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 1. | C CODE WITH COMMENTS FOR TASK 1 (DC Motor) | Y |  |
| 2. | C CODE WITH COMMENTS FOR TASK 2 (Stepper Motor) | Y |  |
| 3. | C CODE WITH COMMENTS FOR TASK 3 (Servo Motor) | Y |  |
| 4. | Schematic Screenshot of TASK1 (DC Motor) | Y |  |
| 5. | Schematic Screenshot of TASK2 (Stepper Motor) | Y |  |
| 6. | Schematic Screenshot of TASK3 (Servo Motor) | Y |  |
| 7 | Flow Diagram of TASK1 | Y |  |
| 8. | Flow Diagram of TASK2 | Y |  |
| 9. | Flow Diagram of TASK3 | Y |  |
| 10. | Snapshot of Connection on Breadboard TASK1/IDE | Y |  |
| 11. | Snapshot of Connection on Breadboard TASK2/IDE | Y |  |
| 12. | Snapshot of Connection on Breadboard TASK3/IDE | Y |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | C CODE WITH COMMENTS FOR TASK 1 ( DC Motor Control ) |  |  |

#include<avr/io.h>

#define F\_CPU 8000000ul

#include<util/delay.h>

#define motor PB0

void ADC\_init(void);

unsigned int ADC\_read(unsigned char);

// ------------------------------------------------

int main(void)

{

uint16\_t value;

DDRB=0xFF;

ADC\_init(); // Initialization of ADC

while(1)

{

value=ADC\_read(0);

PORTB|=1<<motor;

if(value<1)

value=0;

for (int i=0;i<value;i++)

\_delay\_us(1);

PORTB&=~(1<<motor);

for(int i=0;i<1023-value;i++)

\_delay\_us(1);

}

}

//------------------------------------------------

void ADC\_init(void) // Initialization of ADC

{

ADMUX|=(1<<REFS0); // AVcc with external capacitor at AREF

ADCSRA|=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

// Enable ADC and set Prescaler division factor as 128

}

unsigned int ADC\_read(unsigned char ch)

{

ch= ch & 0b00000111; // channel must be b/w 0 to 7

ADMUX |= ch; // selecting channel

ADCSRA|=(1<<ADSC); // start conversion

while(!(ADCSRA & (1<<ADIF))); // waiting for ADIF, conversion complete

ADCSRA|=(1<<ADIF); // clearing of ADIF, it is done by writing 1 to it

return (ADC);

}

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | C CODE WITH COMMENTS FOR TASK 2 ( Stepper Motor Control ) |  |  |

#include<avr/io.h>

#define F\_CPU 8000000ul

#include<util/delay.h>

#define motor PB0

void ADC\_init(void);

unsigned int ADC\_read(unsigned char);

void delay(unsigned int value1)

{

for (int i=0;i<value1;i++)

\_delay\_us(1);

}

// ------------------------------------------------

int main(void)

{

uint16\_t value;

DDRB=0xFF;

ADC\_init(); // Initialization of ADC

while(1)

{

value=ADC\_read(0);

if(value<1)

value=0;

PORTB=0x01;

delay(value);

PORTB=0x02;

delay(value);

PORTB=0x04;

delay(value);

PORTB=0x08;

delay(value);

}

}

//------------------------------------------------

void ADC\_init(void) // Initialization of ADC

{

ADMUX|=(1<<REFS0); // AVcc with external capacitor at AREF

ADCSRA|=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

// Enable ADC and set Prescaler division factor as 128

}

unsigned int ADC\_read(unsigned char ch)

{

ch= ch & 0b00000111; // channel must be b/w 0 to 7

ADMUX |= ch; // selecting channel

ADCSRA|=(1<<ADSC); // start conversion

while(!(ADCSRA & (1<<ADIF))); // waiting for ADIF, conversion complete

ADCSRA|=(1<<ADIF); // clearing of ADIF, it is done by writing 1 to it

return (ADC);

}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. | C CODE WITH COMMENTS FOR TASK 3 ( Servo Motor Control ) |  |  |

#include<avr/io.h>

#define F\_CPU 8000000ul

#include<util/delay.h>

#include <math.h>

#define motor PB0

void ADC\_init(void);

unsigned int ADC\_read(unsigned char);

/\*uint16\_t map1(uint16\_t value,float a,float b,float c, float d)

{

int s=c-a;

float e=d/b;

uint16\_t result=s+(e\*value);

return result;

}\*/

// ------------------------------------------------

int main(void)

{

uint16\_t value;

DDRB=0xFF;

DDRD=0xff;

ADC\_init(); // Initialization of ADc

while(1)

{

value=ADC\_read(0);

if(value<1)

value=1;

for(int j=0;j<20;j++)

{

PORTB|=1<<motor;

for (int i=0;i<value+1000;i++)

\_delay\_us(1);

PORTB&=~(1<<motor);

\_delay\_ms(18);

}

}

}

//------------------------------------------------

void ADC\_init(void) // Initialization of ADC

{

ADMUX|=(1<<REFS0); // AVcc with external capacitor at AREF

ADCSRA|=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

// Enable ADC and set Prescaler division factor as 128

}

unsigned int ADC\_read(unsigned char ch)

{

ch= ch & 0b00000111; // channel must be b/w 0 to 7

ADMUX |= ch; // selecting channel

ADCSRA|=(1<<ADSC); // start conversion

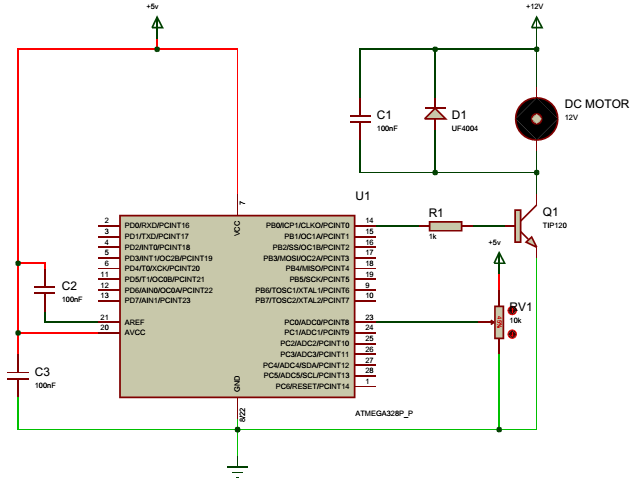
while(!(ADCSRA & (1<<ADIF))); // waiting for ADIF, conversion complete

ADCSRA|=(1<<ADIF); // clearing of ADIF, it is done by writing 1 to it

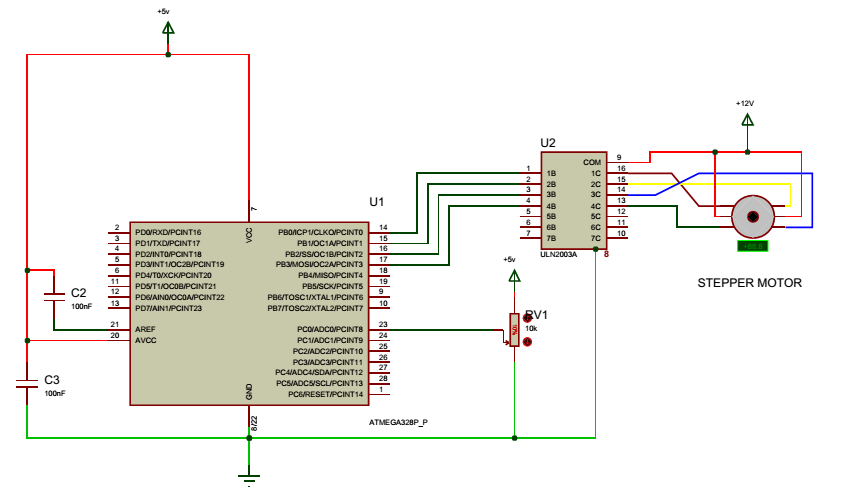
return (ADC);

}

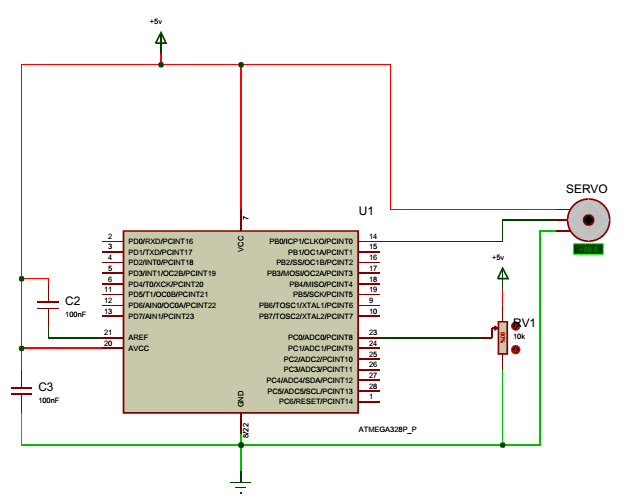
|  |  |  |  |
| --- | --- | --- | --- |
| 4 | Schematic Screenshot of TASK1 (DC Motor) |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Schematic Screenshot of TASK2 (Stepper Motor) |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 6 | Schematic Screenshot of TASK3 (Servo Motor) |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| 7. | Flow Diagram of TASK1 (DC Motor) |  |  |

Initialize IOs

Initialize ADC

Read ADC POT

Start

Run DC Motor

|  |  |  |  |
| --- | --- | --- | --- |
| 8. | Flow Diagram of TASK2 (Stepper Motor) |  |  |

ono

Start

Initialize ADC

Read ADC

Start

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | Flow Diagram of TASK3 (Servo Motor) |  |  |

ono

Start

Initialize ADC

Read ADC

Start

|  |  |  |  |
| --- | --- | --- | --- |
| 10. | SNAPSHOT OF THE COMPUTER SCREEN WITH TERMINAL TASK1 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 11. | SNAPSHOT OF THE COMPUTER SCREEN WITH TERMINAL TASK2 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 12. | SNAPSHOT OF THE COMPUTER SCREEN WITH TERMINAL TASK3 |  |  |

**Student Academic Misconduct Policy**

“This assignment submission is my own, original work”.

Noor Blum