UNIVERSITY OF NORTHERN IOWA

TECH 3157 MICROCONTROLLERS APPLICATIONS SECTION 01

LAB 5: INTERFACING TO ANALOG INPUTS - ANALOG PORT C IN ATmega328P

PROFESSOR: Dr. JIN ZHU

STUDENT: DIDAN JUNQUEIRA RIBEIRO

DECEMBER 4, 2015
CEDAR FALLS, IA

Objectives

1. Understand the operation of the analog port C in ATmega328P
2. Learn to program analog Port C in ATmega328P to obtain analog inputs
3. Learn to interface with phototransistors and temperature sensors.

Equipment

BOE-BOT Shield Kit, Red LED, Pushbutton, 1k and 2k ohm resistor, 0.1uF Capacitor, LM 35, Phototransistor, Buzzer

Procedure and Design

The first lab requirement asked to connect a linear temperature sensor LM35 to the analog port A0 and display the measured temperature in Serial Monitor. The connection diagram used is show below as well the program used to perform that:

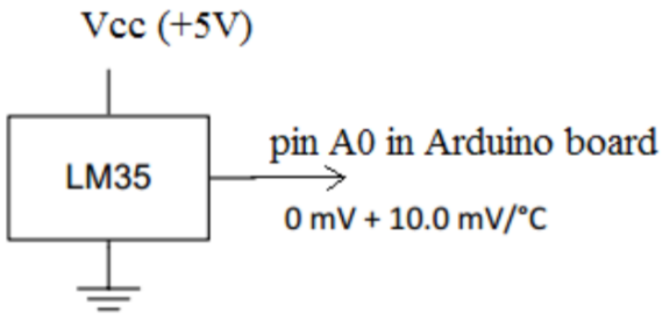
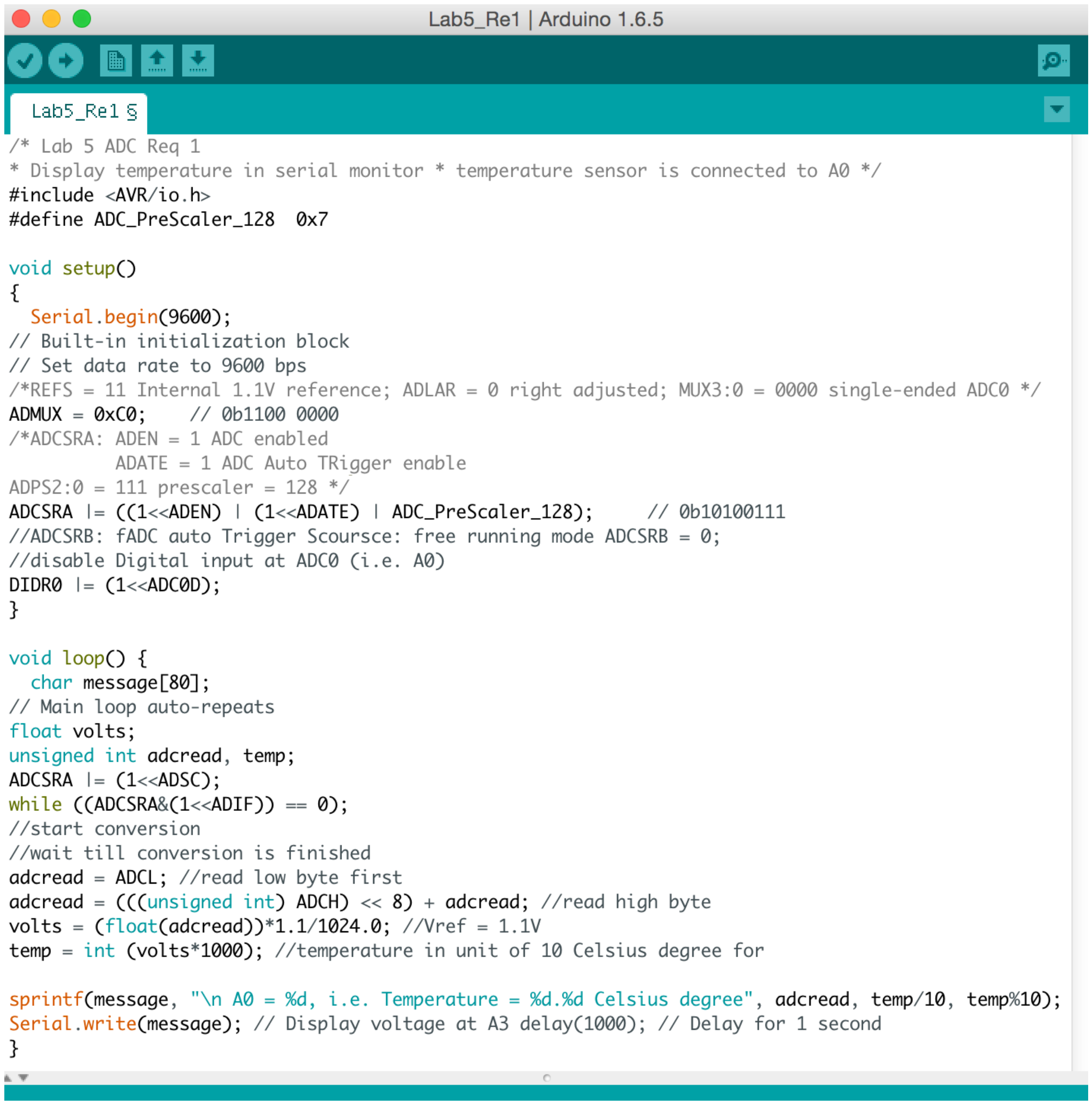


Figure 1: LM35 Connection Diagram

Program used:



```
/* Lab 5 ADC Req 1
* Display temperature in serial monitor * temperature sensor is connected to A0 */
#include <AVR/io.h>
#define ADC_PreScaler_128  0x7

void setup()
{
  Serial.begin(9600);
// Built-in initialization block
// Set data rate to 9600 bps
/*REFS = 11 Internal 1.1V reference; ADLAR = 0 right adjusted; MUX3:0 = 0000 single-ended ADC0 */
ADMUX = 0xC0;    // 0b1100 0000
/*ADCSRA: ADEN = 1 ADC enabled
          ADATE = 1 ADC Auto TRigger enable
ADPS2:0 = 111 prescaler = 128 */
ADCSRA |= ((1<<ADEN) | (1<<ADATE) | ADC_PreScaler_128);     // 0b10100111
//ADCSRB: fADC auto Trigger Scoursce: free running mode ADCSRB = 0;
//disable Digital input at ADC0 (i.e. A0)
DIDR0 |= (1<<ADC0D);
}

void loop() {
  char message[80];
// Main loop auto-repeats
float volts;
unsigned int adcread, temp;
ADCSRA |= (1<<ADSC);
while ((ADCSRA&(1<<ADIF)) == 0);
//start conversion
//wait till conversion is finished
adcread = ADCL; //read low byte first
adcread = (((unsigned int) ADCH) << 8) + adcread; //read high byte
volts = (float(adcread))*1.1/1024.0; //Vref = 1.1V
temp = int (volts*1000); //temperature in unit of 10 Celsius degree for

sprintf(message, "\n A0 = %d, i.e. Temperature = %d.%d Celsius degree", adcread, temp/10, temp%10);
Serial.write(message); // Display voltage at A3 delay(1000); // Delay for 1 second
}
```

Figure 2: Code for Lab requirement 2

The second part of the lab was to observe the analog voltage displayed in the serial monitor based on the example code on the lab 5 manual. Any change was done on that code but through it was possible to observe analog voltage changing when the light level on phototransistor was lighter or darker.

The third part was required to observe the RC time changes. It was achieved by adjusting the threshold and modifying the program to display the returned RC time for each threshold from 2 to 1022 in step of 2. The The RC circuit using a phototransistor is show below as well the code used to perform the requirement:

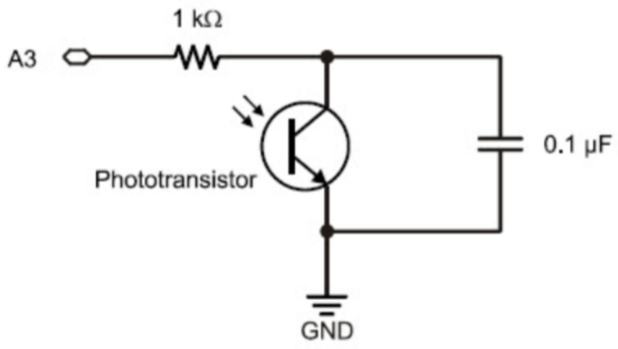


Figure 3: RC circuit using a phototransistor

The code used was:

```
/* Lab 5 req 3

* Robotics with the BOE Shield - Phototransistor connection using RC circuit

* * Display the discharge time in microseconds

* in the serial monitor.

*/

#include <AVR/io.h>
```

```
#define ADC_PreScaler_128  0x7

#define ADC3 3

#define ADC4 4

unsigned int voltsLevel(char ADCpin);

unsigned long rcTime(char ADCpin, int  threshold);


void setup()

{

  Serial.begin(9600);

// Built-in initialization block

// Set data rate to 9600 bps

/*REFS = 01 Internal 5V reference; ADLAR = 0 right adjusted;

MUX3:0 = 0011 single-ended ADC3 (pin A3 in Ariduino board) */

 ADMUX = 0x40 | 0x03;

 /*ADCSRA: ADEN = 1 ADC enabled

            ADATE = 1 ADC Auto TRigger enable

ADPS2:0 = 111 prescaler = 128 */

ADCSRA |= ((1<<ADEN) | (1<<ADATE) | ADC_PreScaler_128);

ADCSRB = 0;

 char message[80];

  float volts;

  unsigned int adcread[100];
  int threshold;
            for (threshold = 2; threshold < 1023; threshold +=2)
             {
            // Main loop auto-repeats
            DDRC |= 0x08; //set pin A3 (portC.3) as an output
            PORTC |= 0x08; //output HIGH at pin A3
            delay(1); //hold 5V for 1 ms to allow charging the capacitor
            DDRC &= ~0x08; //set pin A3 as an input
            PORTC &= ~0x08; //disconnect the pullup-resistor to allow discharge
            unsigned long time = micros();
            while (voltsLevel(ADC3)>threshold); //wait till the read from ADC3 is less than 10
            time = micros() - time;
            sprintf(message, "\n rctime = %lu", time);
            Serial.write(message);
            sprintf(message, "\n threshold = %d", threshold);
            Serial.write(message);
             }
}

void loop()
{
}
```

```
unsigned int voltsLevel(char ADCpin)
{
    unsigned int adcread;
    //disable Digital input at ADCpin
    DIDR0 |= (1<<ADCpin);
    ADCSRA |= (1<<ADSC);
    while ((ADCSRA&(1<<ADIF)) == 0);
    adcread = ADCL;
    adcread = (((int) ADCH) << 8) + adcread; //read high byte
    return adcread;
} // End of lab requirement 3
```

The last part asked to use the same phototransistor RC circuits for both side of robot and guide its movement according to the light level. The circuit connection diagram for that and code is shown below:

```
/* Lab 5 last req
* Robotics with the BOE Shield - Phototransistor connection using RC circuit
* * Display the discharge time in microseconds
* in the serial monitor.
*/
#include <AVR/io.h>
```

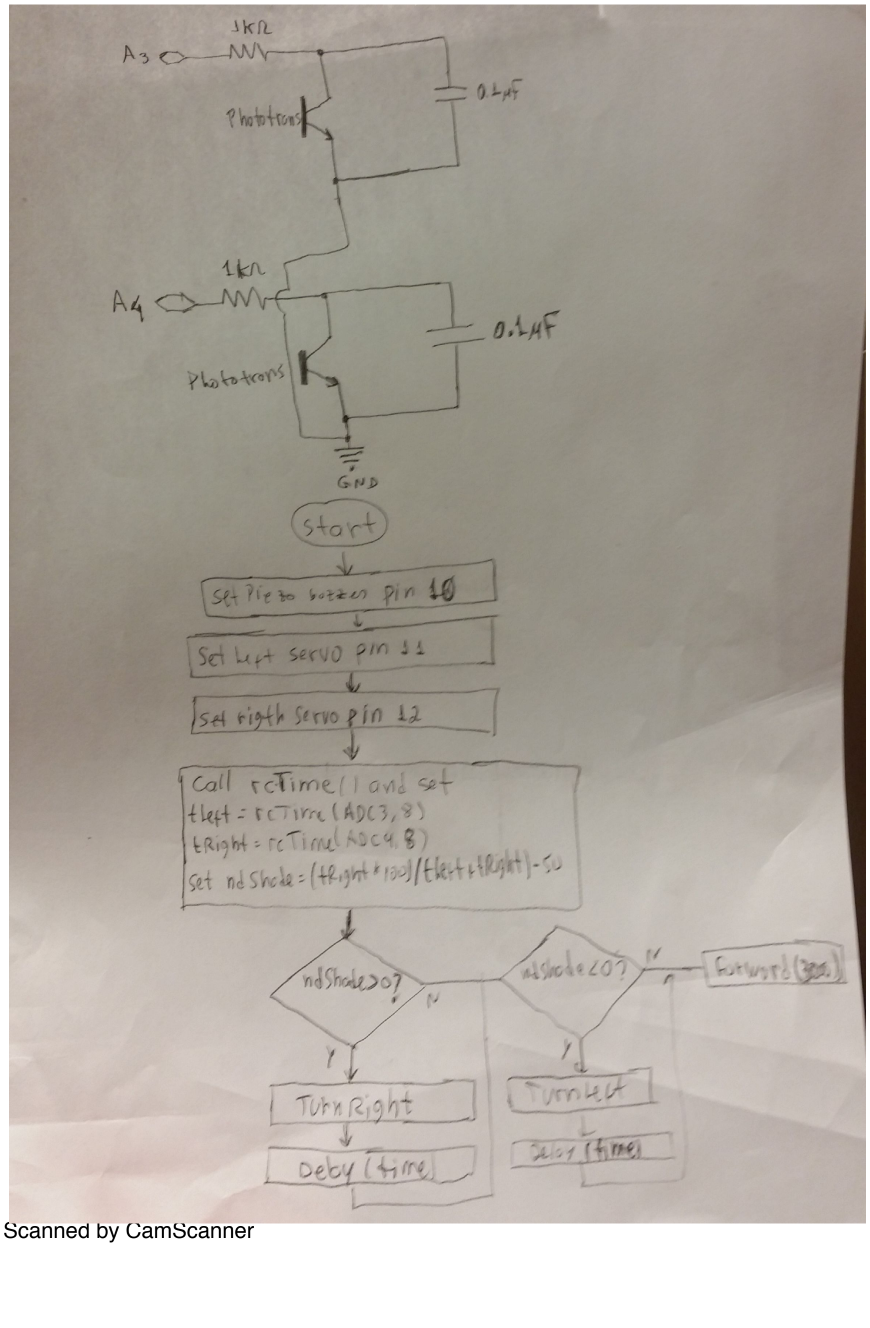
```
#define ADC_PreScaler_128  0x7

#define ADC3 3

#define ADC4 4


unsigned long rcTime(char ADCpin, int  threshold);

unsigned int voltsLevel(char ADCpin);

#include <Servo.h>      // Include servo library


void forward  (unsigned int time);    //full speed forward for a duration of time (ms)

void turnRight(unsigned int time) // Turn right

void turn Left(unsigned int time) // Turn left

void disableServos();   // Halt servo signals


Servo servoLeft;        // Declare left and right servos

Servo servoRight;


void setup()

{

  tone(10, 3000, 1000);     // Play tone for 1 second

  delay(1000);              // Delay to finish the tone

  servoLeft.attach(11);     // Attach left signal to pin 11

  servoRight.attach(12);    // Attach right signal to pin 12
  Serial.begin(9600);
// Built-in initialization block
// Set data rate to 9600 bps
/*REFS = 01 Internal 5V reference; ADLAR = 0 right adjusted;
MUX3:0 = 0011 single-ended ADC3 (pin A3 in Ariduino board) */
//ADMUX = 0x40 | 0x03;
            /*ADCSRA: ADEN = 1 ADC enabled
            ADATE = 1 ADC Auto TRigger enable
            ADPS2:0 = 111 prescaler = 128 */
    ADCSRA |= ((1<<ADEN) | (1<<ADATE) | ADC_PreScaler_128);
    //ADCSRB: fADC auto Trigger Scoursce: free running mode
    ADCSRB = 0;
    //disable Digital input at ADC3
    //DIDR0 |= (1<<ADC3D);
}

void loop()
{
 char message[80];
 unsigned long tLeft;
 unsigned long tRight;
 char ndShade;            // Zero Normalized differential shade
```

```
tLeft = rcTime(ADC3, 8) ;    //wait till the read from ADC3

tRight = rcTime(ADC4, 8);   //wait till the read from ADC4

ndShade = (tRight*100) / (tLeft + tRight) - 50;


sprintf(message, "\n   ndShade = %d ", ndShade);

Serial.write(message);


if (ndShade > 0)

{

  servoLeft.writeMicroseconds(1700 - (ndShade*5) );

  servoRight.writeMicroseconds(1300);

}

else if (ndShade < 0)

{

  servoLeft.writeMicroseconds (1700);

  servoRight.writeMicroseconds(1300 - (ndShade*5) );

}

else

{

  forward(300);

}
```

```
}

unsigned int voltsLevel(char ADCpin)

{

  unsigned int adcread;

  //disable Digital input at ADCpin

  DIDR0 |= (1<<ADCpin);

  ADCSRA |= (1<<ADSC);

  while ((ADCSRA&(1<<ADIF)) == 0);

  adcread = ADCL;

  adcread = (((int) ADCH) << 8) + adcread; //read high byte

  return adcread;

}


unsigned long rcTime(char ADCpin, int  threshold)

{

  char message[80];

  ADMUX = 0x40 | ADCpin;

  DDRC |= (1<<ADCpin);       //0x08; //set pin A3 (portC.3) as an output

  PORTC |= (1<<ADCpin);     //0x08; //output HIGH at pin A3

  delay(1); //hold 5V for 1 ms to allow charging the capacitor

  DDRC &= ~(1<<ADCpin);     // 0x08 set pin A3 as an input

  PORTC &= ~(1<<ADCpin);    //0x08 disconnect the pullup-resistor to allow discharge
  unsigned long time = micros();

  while (voltsLevel(ADCpin>threshold)); //wait till the read from ADC3

  time = micros() - time;

  return time;

}


void forward (unsigned int time)   // Full-speed Forward function

{

  servoLeft.writeMicroseconds(1625);

  servoRight.writeMicroseconds(1300);

  delay(time);    // Maneuver for time ms

  }

void turnRight(unsigned int time)   // Right turn function

{

servoLeft.writeMicroseconds(1700);

servoRight.writeMicroseconds(1700);

delay(time);     // Maneuver for time ms

}

void turnLeft(unsigned int time)  // Left turn function

{

servoLeft.writeMicroseconds(1300);

servoRight.writeMicroseconds(1300);
```

delay(time); // Maneuver for time ms } // End lab req 4

Diagram and flowchart:



Scanned by CamScanner

Figure 4: Circuit diagram and flowchart

Conclusion

All the lab requirements was achieved. Following the hints in the lab 5 manual and using some programming background was possible to implement the lab requirement 5 and programming a robot guided by light using phototransistors.