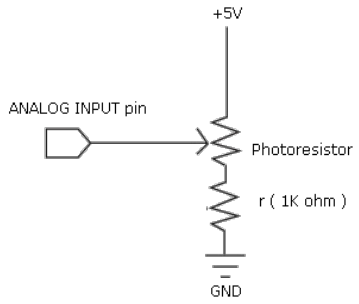
ΠΑΝ ΔΥΤ. ΜΑΚΕΔΟΝΙΑΣ **ΤΜΗΜΑ ΜΗΧ. ΠΛΗΡΟΦΟΡΙΚΗΣ κ ΤΑΠ.**
ΧΕΙΜ. ΕΞΑΜ. 2014-2015 ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ
Διδάσκων: Κοντογιάννης Σωτήριος

ΕΡΓΑΣΤΗΡΙΟ 6

ΑΣΚΗΣΗ 1: Να δημιουργήσετε ένα πρόγραμμα σε C (χωρίς χρήση της συνάρτησης analogRead) και assembly το οποίο θα χρησιμοποιεί το παρακάτω κύκλωμα και θα διαβάζει από την αναλογική είσοδο 3 και θα εμφανίζει στη σειριακή έξοδο την τιμή του αναλογικού αισθητήρα (photoresistor):



Υπόδειξη 1:

```
uint16_t a;
void setup()
{
  InitADC();
  Serial.begin(9600);
}

void loop()
{
  a=ReadADC(3);
  Serial.println(a);
}

uint16_t ReadADC(uint8_t ch)
{
   //Select ADC Channel ch must be 0-7
   ch=ch&0b00000111;
   ADMUX|=ch;
   //Start Single conversion
   ADCSRA|=(1<<ADSC);
   //Wait for conversion to complete
   while(!(ADCSRA & (1<<ADIF)));
   //Clear ADIF by writing one to it
   ADCSRA|=(1<<ADIF);
   return(ADC);
}

void InitADC()
{
ADMUX=(1<<REFS0);                              // For Aref=AVcc;
ADCSRA=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0); //Rrescalar div factor =128
}
```

ΑΣΚΗΣΗ 2: Να δημιουργήσετε ένα πρόγραμμα σε assembly που προσομοιώνει τη λειτουργία ενός αερόφωνου χρησιμοποιώντας το κύκλωμα της άσκησης 1 συνδεδεμένο στην αναλογική είσοδο 3 και ένα buzzer συνδεδεμένο στην ψηφιακή έξοδο 11.

Υπόδειξη 2:

```
/*Automatically subtract 0x20 from I/O addresses
*/
#define __SFR_OFFSET 0
#include "adc1.h"
#include <avr/io.h>
#include <avr/interrupt.h>
.section .data
.global adval
adval: .byte 0x00
.section .text
.global aetherphone
aetherphone:
reset:
;Clear analog input pins with no pull ups (PORTC)
   CLR r16
   OUT  DDRC, r16       ;all pins input
   OUT PORTC,r16        ;no pull ups, Tri-State (Hi-Z 10K optimal impedance)
  CBI PORTB,3 ;PIN 11
  SBI DDRB,3 ;; ALL DDRB
rloop:
  // ldi        r16, (1<<ADEN) | (1<<ADSC) | (1<<ADLAR) | (0<<MUX3) | (1<<MUX2) | (1<<MUX1) | (1<<MUX0)
   LDI r16,0b11000111 ;Enable start conversrion and set
   ;prescaler to 128
   STS ADCSRA,r16
   LDI   r16,0b01000011  ;SELECT ADC in 3 PORTC 3
   STS  ADMUX,r16

wait:
    LDS r16,ADCSRA
    ANDI r16,0b00010000
    BREQ wait
    LDS   r16,ADCL         ;MUST READ ADCL BEFORE ADCH
    LDS    r17,ADCH        ;REQUIRED, THOUGH NOT USED
    STS   adval,r16
    RCALL pause
    RCALL sound
    RJMP rloop
ret
sound:
  LDI r20,_BV(3)
  IN r18,PORTB
  EOR r18,r20
  OUT PORTB, r18
ret
pause:
  DEC r16
  BRNE pause
ret
.end
```

ΑΣΚΗΣΗ 3: Να δημιουργήσετε πρόγραμα σε C που θα χρησιμοποιεί την αναλογική θύρα 3 του κυκλώματος της άσκησης 1 σε κατάσταση free running και που θα διαβάζει μετά τις 255 τις 10 αναλογικές τιμές και θα υπολογίζει το μέσο όρο τους χρησιμοποιώντας την ISR(ADC_vect).

Υπόδειξη 3:

```
int sendStatus = 0; // send status
int startDelay = 0;
byte valueBin[1024]; // value bins

void setup() {
TIMSK0 = 0x00; // disable timer (causes anoying interrupts)
DIDR0 = 0x3F; // digital inputs disabled
ADMUX = 0xC3; // measuring on ADC3, left adjust, internal 1.1V ref
ADCSRA = 0xAC; // AD-converter on, interrupt enabled, prescaler = 128
ADCSRB = 0x40; // AD channels MUX on, free running mode
bitWrite(ADCSRA, 6, 1); // Start the conversion by setting bit 6 (=ADSC) in ADCSRA
sei(); // Set global interrupt flag
for (int i=0; i<=1023; i++) { // clear bins
valueBin[i] = 0;
}
Serial.begin(9600);
Serial.println("Start");
}

void loop() {
if (sendStatus == 1) {
for (int i=0; i<=1023; i++) { // output bin values
Serial.print(i);
Serial.print("\t");
Serial.println(valueBin[i]);
}
Serial.println("Done");
sendStatus = 2;
}
}
        /*** Interrupt routine ADC ready ***/
        ISR(ADC_vect) {
        int aval = ADCL; // store lower byte ADC
        aval += ADCH << 8; // store higher bytes ADC
        if (sendStatus == 0) {
        if (startDelay < 10000) { // do nothing the first x samples
        startDelay++;
        }
        else {
        valueBin[aval] += 1; // increase value bin
        if (valueBin[aval] == 255) { // stop if a bin is full
        sendStatus = 1;
        }
        }
        }
        //Changing analog pins
        //if (ADMUX == 0xC3) {
        //ADMUX = 0xC5;}
        //else {
        //ADMUX = 0xC3;}
        }
```