



Universidad
del Caribe

2000

CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

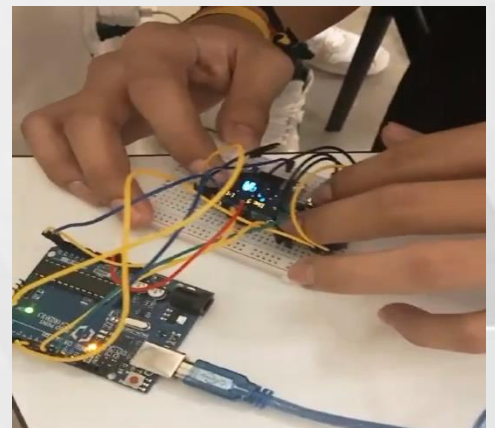
Diseño y organización de computadoras

Proyecto final

Ingeniería en datos e inteligencia organizacional

Integrantes:

- ✓ Amauri Alejandro Ventura Garcia
- ✓ Braiam Augusto Hernandez Cen
- ✓ Joab ahiezer xool Dzul
- ✓ Victor Andres Gomez Gonzales
- ✓ Jared Alexander Gongora Tuyub
- ✓ Alexis de la cruz calderón
- ✓ Jesus Adrian Canche Sandoval



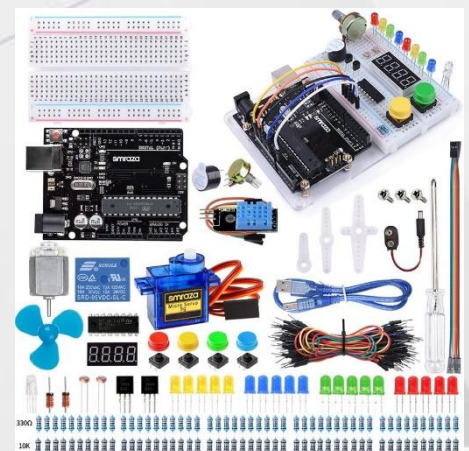
Elaboración

Nuestro proyecto lo elaboramos con una base a programar en lo que es el Arduino un videojuego a base también de una pantalla mini oled para que se refleje todo el programa a través de este, tuvimos que hacer un código que este refleje la base de nuestro videojuego ya una vez que el código lo pudimos completar utilizamos una aplicación para que el Arduino pueda reconocer el código y ejecutarlo la aplicación que utilizamos para que el código fuera reconocido fue la de Arduino IDE que nos facilitó ese proceso.

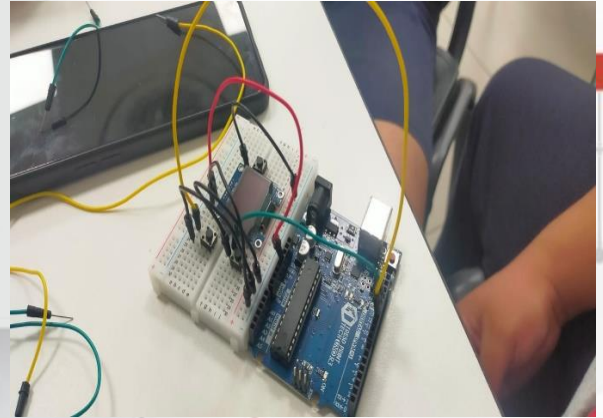


Este fue nuestro kit de armado para nuestro proyecto y poder hacer la elaboración adecuadamente, realmente lo más difícil de nuestro proyecto fue el armado ya que al principio no daba lo

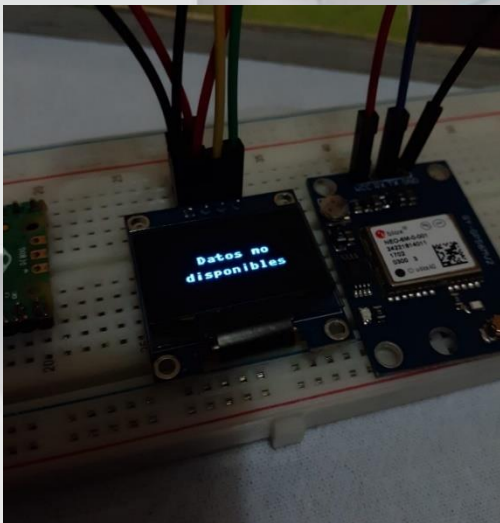
que queríamos , pero después de intentarlo muchas veces quedó bien en lo que correspondía al armado.



Después del armado de lo que fueron los cables pudimos ya poner nuestra pantalla oled y ver si realmente el código funcionaba y al principio como siempre no nos estaba funcionando lo que es la pantalla pero era porque el conectado nos salió mal en dos cables pero una vez que ya pudimos conectarlo nos pudo dar la imagen

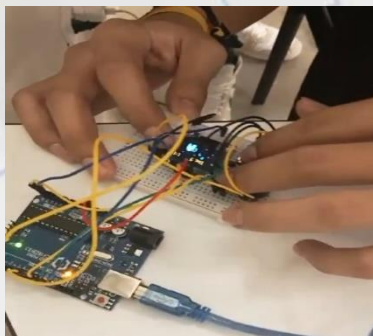


Como podemos ver en la imagen en la izquierda es cuando nos mandaba error de lo que era el juego pensabamos que era el codigo pero no, fue la mala gestion de de lo que son los conectores.



Como podemos ver en la imagen en la izquierda es cuando nos mandaba error de lo que era el juego pensabamos que era el codigo pero no, fue la mala gestion de de lo que son los conectores ya que la segunda imagen es cuando ya coregimos nos dio por fin la imagen que es la foto de la derecha.

Finalmente cuando se realizo la revision del codigo, el armado y distribución de cables, se pudo apreciar el resultado final del proyecto que viene siendo el juego de star wars aplicado en un arduino. Con esto nos dimos cuenta de que no es sencillo realizar proyectos de este tipo ya que se necesita dedicacion, conocimientos previos de programacion y sobre todo paciencia tanto para el armado del mismo como verificar y corregir los errores que puede salir durante el proyecto.



Herramientas Y Costo

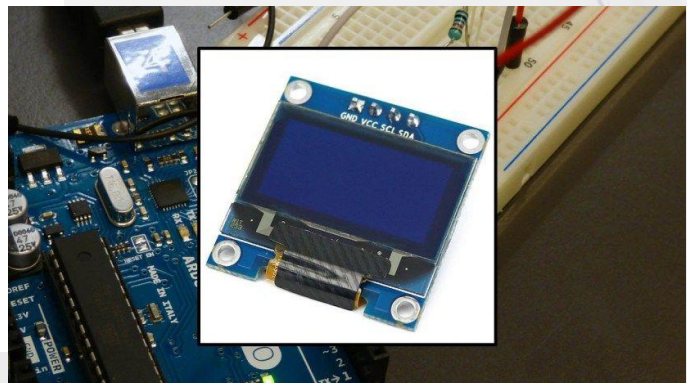
Para la elaboracion del proyecto del arduino se necesito las siguientes materiales con sus respectivos costos:

Cada uno de estos productos fue adquirido en tienda en linea por medio de mercado libre por proveedores de confianza y calificados para la entrega.

1. Kit Arduino: \$711.80 pesos.



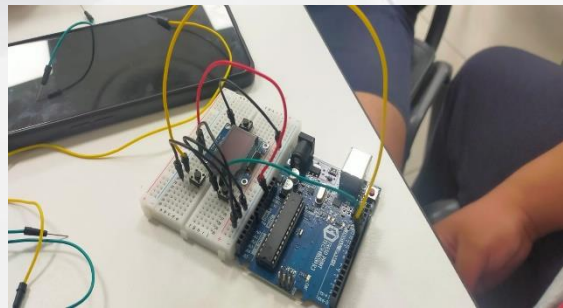
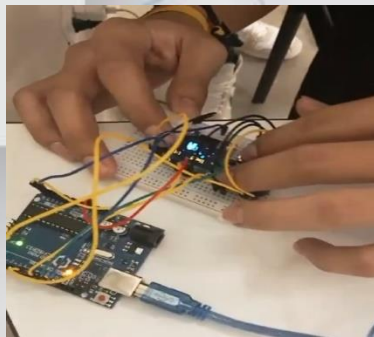
2. Pantalla Oled: \$65.90 pesos.



Conclusión

Como puede ver este es el código que utilizamos ya que verdaderamente es un poco largo, pero gracias a eso nuestro código funciona a la perfección cumple lo que es su propósito y realmente esto es lo que hace que el Arduino lo reconozca y pueda ejecutar en la pantalla y correr el juego.

Para terminar, este proyecto nos enseñó aprendizajes y conocimientos para el futuro, ya que logramos no solamente ensamblar, si no también funcionar y de manera correcta para el proyecto, con esto se pudo aprender que para realizar proyectos y trabajos así se necesita conocimientos previos, saber programación y sobre todo paciencia para poder no solamente ensamblar, si no poder revisar y corregir los errores que puede a ver en el futuro y tener un proyecto o trabajo excelente.



Codigo

El código que utilizamos no fuera mente sencilla elaborarlo ya que cuando lo hicimos igual nos arrojaba errores igual, al final de todo pudimos corregirlo y poder avanzar que literal fue nuestra base para poder hacer este proyecto.

```
/******
```

This is an example for our Monochrome OLEDs based on SSD1306 drivers

Pick one up today in the adafruit shop!

-----> http://www.adafruit.com/category/63_98

This example is for a 128x32 size display using I2C to communicate

3 pins are required to interface (2 I2C and one reset)

Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing
products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.

BSD license, check license.txt for more information

All text above, and the splash screen must be included in any redistribution

```
*****/
```

```
#include <SPI.h>
```

```
#include <Wire.h>
```

```
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
#include <Fonts/FreeSans9pt7b.h>
#include <Fonts/FreeSans12pt7b.h>
```

```
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);
```

```
const int c = 261;
const int d = 294;
const int e = 329;
const int f = 349;
const int g = 391;
const int gS = 415;
const int a = 440;
const int aS = 455;
const int b = 466;
const int cH = 523;
const int cSH = 554;
const int dH = 587;
const int dSH = 622;
const int eH = 659;
const int fH = 698;
const int fSH = 740;
const int gH = 784;
const int gSH = 830;
const int aH = 880;
```

```
const unsigned char PROGMEM dioda16 [] = {
```

```
0x00, 0x00, 0x00, 0x00, 0x1C, 0x00, 0x3F, 0xF0, 0x3C, 0x00, 0x3C, 0x00, 0xFF, 0x00, 0x7F, 0xFF,
0x7F, 0xFF, 0xFF, 0x00, 0x3C, 0x00, 0x3C, 0x00, 0x1F, 0xF0, 0x1C, 0x00, 0x00, 0x00, 0x00
```



```
};
```

```
const unsigned char PROGMEM storm [] = {
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x7F, 0xFE, 0x00, 0x00, 0x00, 0x07, 0x80, 0x01, 0xE0, 0x00, 0x00, 0x0C,  
0x00, 0x00, 0x20, 0x00, 0x00, 0x18, 0x00, 0x00, 0x18, 0x00, 0x00, 0x30, 0x00, 0x00, 0x04, 0x00,  
0x00, 0x20, 0x00, 0x00, 0x04, 0x00, 0x00, 0x20, 0x00, 0x00, 0x04, 0x00, 0x00, 0x60, 0x00, 0x00,  
0x02, 0x00, 0x00, 0x40, 0x00, 0x00, 0x02, 0x00, 0x00, 0x40, 0x00, 0x00, 0x01, 0x00, 0x00, 0x40,  
0x00, 0x00, 0x01, 0x00, 0x00, 0x40, 0x00, 0x00, 0x01, 0x00, 0x00, 0x7F, 0xE0, 0x00, 0x01, 0x00,  
0x00, 0x7F, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x7F, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0xD7, 0xFF, 0xFF,  
0xE1, 0x00, 0x01, 0xBF, 0xFC, 0x1F, 0xFA, 0x80, 0x01, 0xBF, 0xF1, 0xCF, 0xFA, 0x80, 0x01, 0x3F,  
0xC2, 0x37, 0xF7, 0x80, 0x01, 0xEF, 0x9C, 0x01, 0xE7, 0xC0, 0x01, 0xE0, 0x70, 0x06, 0x06, 0x80,  
0x01, 0xE0, 0xC0, 0x03, 0x06, 0x80, 0x01, 0xFF, 0x80, 0x01, 0xFF, 0x80, 0x01, 0xF8, 0x00, 0x00,  
0x1D, 0xC0, 0x03, 0x70, 0x00, 0x80, 0x0C, 0x60, 0x05, 0xB0, 0x07, 0xF0, 0x08, 0x90, 0x09, 0x10,  
0x1F, 0xF8, 0x09, 0xD0, 0x0B, 0x90, 0x1F, 0x7C, 0x03, 0xF0, 0x0F, 0xC0, 0xFC, 0x0F, 0x07, 0x90,  
0x0D, 0x43, 0xC0, 0x03, 0x07, 0x90, 0x05, 0x64, 0x00, 0x00, 0xCF, 0x10, 0x07, 0xFC, 0x00, 0x00,  
0x26, 0x10, 0x01, 0x80, 0x00, 0x00, 0x10, 0x20, 0x01, 0x00, 0x00, 0x00, 0x0E, 0x40, 0x01, 0x80,  
0x07, 0xF0, 0x01, 0x80, 0x00, 0x80, 0x07, 0xC8, 0x00, 0x80, 0x00, 0x80, 0x0B, 0xE8, 0x00, 0x80,  
0x00, 0x87, 0x97, 0xE9, 0xE0, 0x80, 0x00, 0x87, 0xDF, 0xEF, 0xA0, 0x80, 0x00, 0x4B, 0xFF, 0xFF,  
0xA0, 0x80, 0x00, 0x6B, 0xDF, 0xFB, 0xA3, 0x00, 0x00, 0x24, 0x97, 0xE8, 0x24, 0x00, 0x00, 0x1E,  
0x1F, 0xC0, 0x2C, 0x00, 0x00, 0x07, 0xF8, 0x1F, 0xF0, 0x00, 0x00, 0x00, 0x0F, 0xF8, 0x00, 0x00  
};
```

```
void setup() {
```

```
pinMode(3,INPUT_PULLUP);
```

```
pinMode(12,INPUT_PULLUP);
```

```
pinMode(11,INPUT_PULLUP);
```

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
```



```
display.display();

display.clearDisplay();

display.setTextSize(0);

display.drawBitmap(6, 11, storm, 48, 48, 1);

display.setFont(&FreeSans9pt7b);

display.setTextColor(WHITE);

display.setCursor(65, 14);

display.println("xWing");

display.setFont();

display.setCursor(65, 17);

display.setTextSize(0);

display.println("vs");

display.setCursor(0, 0);

float voltaza = readVcc() / 1000;

display.println(voltaza);

display.setFont(&FreeSans9pt7b);

display.setCursor(65, 39);

display.println("Death");

display.setFont();

display.setCursor(65, 42);

display.println("star ");

display.setTextSize(0);


display.setCursor(65, 55);


display.println("by Danko");


display.setCursor(65, 20);
```

```
display.display();
```

```
display.setFont();
```

```
beep(a, 500);
```

```
beep(a, 500);
```

```
beep(a, 500);
```

```
beep(f, 350);
```

```
beep(cH, 150);
```

```
beep(a, 500);
```

```
beep(f, 350);
```

```
beep(cH, 150);
```

```
beep(a, 650);
```

```
delay(500);
```

```
delay(500);
```

```
}
```

```
int metx=0;
```

```
int mety=0;
```

```
int postoji=0;
```

```
int nep=8;
```

```
int smjer=0;
```

```
int go=0;
```

```
int rx=95;
```

```
int ry=0;
```

```
int rx2=95;  
int ry2=0;  
int rx3=95;  
int ry3=0;  
int bodovi=0;
```

```
int brzina=3; //speed of bullet  
int bkugle=1;  
int najmanja=600;  
int najveca=1200;  
int promjer=10;
```

```
int rx4=95;  
int ry4=0;  
int zivoti=5;  
int poc=0;  
int ispaljeno=0;  
int nivo=1;  
int centar=95;  
unsigned long pocetno=0;  
unsigned long odabrano=0;  
unsigned long trenutno=0;  
unsigned long nivovrije=0;  
int poz=30;  
void loop() {
```

```
    if(go==0){  
        display.clearDisplay();
```

```
display.drawPixel(50,30,1);
display.drawPixel(30,17,1);
display.drawPixel(60,18,1);
display.drawPixel(55,16,1);
display.drawPixel(25,43,1);
display.drawPixel(100,43,1);
display.drawPixel(117,52,1);
display.drawPixel(14,49,1);
display.drawPixel(24,24,1);
display.drawPixel(78,36,1);
display.drawPixel(80,57,1);
display.drawPixel(107,11,1);
display.drawPixel(150,11,1);
display.drawPixel(5,5,1);
display.drawPixel(8,7,1);
display.drawPixel(70,12,1);
display.drawPixel(10,56,1);
display.drawPixel(70,25,1);
```

```
if(poc==0){
pocetno=millis();
odabrano= random(400,1200);
poc=1;
}
trenutno=millis();
```



```
//nivoi
```

```
if((trenutno-nivovrije)>50000)
```

```
{
```

```
    nivovrije=trenutno;
```

```
    nivo=nivo+1;
```

```
brzina=brzina+1; //brizna neprijateljevog metka
```

```
if ( nivo % 2 == 0 )
```

```
{
```

```
    bkugle=bkugle+1;
```

```
    promjer=promjer-1;
```

```
}
```

```
najmanja=najmanja-50;
```

```
najveca=najveca-50;
```

```
}
```

```
if((odabrano+pocetno)<trenutno)
```

```
{
```

```
    poc=0;
```

```
    ispaljeno=ispaljeno+1;
```

```
    if(ispaljeno==1)
```

```
    {rx=95;
```

```
    ry=nep;}
```

```
    if(ispaljeno==2){
```

```
        rx2=95;
```

```
ry2=nep;}  
if(ispaljeno==3)  
{  
    rx3=95;  
    ry3=nep;  
}
```

```
if(ispaljeno==4){  
    rx4=95;  
    ry4=nep;  
  
}
```

```
}
```

```
if(ispaljeno>0)  
{  
    display.drawCircle(rx,ry,2,1);  
    rx=rx-brzina;}  

```

```
if(ispaljeno>1)  
{  
    display.drawCircle(rx2,ry2,1,1);  
    rx2=rx2-brzina;}  

```

```
if(ispaljeno>2)
{
display.drawCircle(rx3,ry3,4,1);
rx3=rx3-brzina;}
```

```
if(ispaljeno>3)
{
display.drawCircle(rx4,ry4,2,1);
rx4=rx4-brzina;}
```

```
if(digitalRead(12)==0 && poz>=2){
poz=poz-2;}
```

```
if(digitalRead(11)==0&& poz<=46){
poz=poz+2;}
```

```
if(digitalRead(3)==0 && postoji==0)
{
postoji=1;
metx=6;
mety=poz+8;
tone(9,1200,20);
```

```
}
```

```
if(postoji==1)
```

```
{
```

```
metx=metx+8 ;
```

```
display.drawLine(metx,mety,metx+4,mety,1);  
}
```

```
display.drawBitmap(4, poz,dioda16, 16,16, 1);  
display.fillCircle(centar,nep,promjer,1);  
display.fillCircle(centar+2,nep+3,promjer/3,0);
```

```
display.setTextSize(1);  
display.setTextColor(WHITE);  
display.setCursor(33,57);  
display.println("score:");  
    display.setCursor(68,57);  
display.println(bodovi);
```

```
display.setCursor(33,0);  
display.println("lives:");  
    display.setCursor(68,0);  
display.println(zivoti);
```

```
display.setCursor(110,0);  
display.println("L:");
```

```
display.setCursor(122,0);  
display.println(nivo);
```

```
display.setCursor(108,57);  
display.println(trenutno/1000);
```



```
display.display();
```

```
if(metx>128)postoji=0;
```

```
if(smjer==0){  
    nep=nep+bkugle;}  
else  
    {nep=nep-bkugle;}
```

```
if(nep>=(64-promjer))  
    smjer=1;  
if(nep<=promjer)  
    smjer=0;
```

```
if(mety>=nep-promjer && mety<=nep+promjer)  
if(metx>(centar-promjer)&&metx<(centar+promjer))  
{  
    metx=-20;  
    tone(9,500,20);  
    bodovi=bodovi+1;  
    postoji=0;  
}
```

```
int pozicija=poz+8;  
    if(ry>=pozicija-8 && ry<=pozicija+8)  
    if(rx<12&&rx>4)  
    {
```

```
rx=95;  
ry=-50;  
tone(9,100,100);  
zivoti=zivoti-1;  
}
```

```
    if(ry2>=pozicija-8 && ry2<=pozicija+8)  
    if(rx2<12&&rx2>4)  
    {  
        rx2=-50;  
        ry2=-50;  
        tone(9,100,100);  
        zivoti=zivoti-1;  
    }
```

```
    if(ry3>=pozicija-8 && ry3<=pozicija+8)  
    if(rx3<12&&rx3>4)  
    {  
        rx3=-50;  
        ry3=-50;  
        tone(9,100,100);  
        zivoti=zivoti-1;  
    }
```

```
    if(ry4>=pozicija-8 && ry4<=pozicija+8)  
    if(rx4<12&&rx4>4)  
    {  
        rx4=200;  
        ry4=-50;
```

```
    ispaljeno=0;

    tone(9,100,100);
    zivoti=zivoti-1;
}
```

```
if(rx4<1){
    ispaljeno=0;
    rx4=200;}


```

```
if(zivoti==0)
go=1;
}
```

```
if(go==1)
{
    if(zivoti==0){
        tone(9,200,300);
        delay(300);
        tone(9,250,200);
        delay(200);
        tone(9,300,300);
        delay(300);
        zivoti=5;
    }

    display.clearDisplay();

    display.setFont();

    display.setTextSize(2);

    display.setTextColor(WHITE);

    display.setCursor(7,10);
}
```

```
display.println("GAME OVER!");  
  
display.setTextSize(1);  
display.setCursor(7,30);  
display.println("score:");  
display.setCursor(44,30);  
display.println(bodovi);  
display.setCursor(7,40);  
display.println("level:");  
display.setCursor(44,40);  
display.println(nivo);  
display.setCursor(7,50);  
display.println("time(s):");  
display.setCursor(60,50);
```

```
display.println(trenutno/1000);  
display.display();
```

```
if(digitalRead(3)==0)  
{  
    tone(9,280,300);  
    delay(300);  
    tone(9,250,200);  
    delay(200);  
    tone(9,370,300);  
    delay(300);  
    ponovo();  
}  
}  
}
```



```
void ponovo()
```

```
{
```

```
    metx=0;
```

```
    mety=0;
```

```
    postoji=0;
```

```
    nep=8;
```

```
    smjer=0;
```

```
    go=0;
```

```
    rx=95;
```

```
    ry=0;
```

```
    rx2=95;
```

```
    ry2=0;
```

```
    rx3=95;
```

```
    ry3=0;
```

```
    bodovi=0;
```

```
    brzina=3; //brizna neprijateljevog metka
```

```
    bkugle=1;
```

```
    najmanja=600;
```

```
    najveca=1200;
```

```
    promjer=12;
```

```
    rx4=95;
```

```
    ry4=0;
```

```
    zivoti=5;
```

```
    poc=0;
```

```
    ispaljeno=0;
```

```
    nivo=1;
```

```
pocetno=0;
odabrano=0;
trenutno=0;
nivovrije=0;
```

```
}
```

```
long readVcc() {
    // Read 1.1V reference against AVcc
    // set the reference to Vcc and the measurement to the internal 1.1V reference
    #if defined(_AVR_ATmega32U4) || defined(AVR_ATmega1280) || defined(AVR_ATmega2560_)
        ADMUX = _BV(REFS0) | _BV(MUX4) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
    #elif defined (_AVR_ATtiny24) || defined(AVR_ATtiny44) || defined(AVR_ATtiny84_)
        ADMUX = _BV(MUX5) | _BV(MUX0);
    #elif defined (_AVR_ATtiny25) || defined(AVR_ATtiny45) || defined(AVR_ATtiny85_)
        ADMUX = _BV(MUX3) | _BV(MUX2);
    #else
        ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
    #endif

    delay(2); // Wait for Vref to settle
    ADCSRA |= _BV(ADSC); // Start conversion
    while (bit_is_set(ADCSRA,ADSC)); // measuring

    uint8_t low  = ADCL; // must read ADCL first - it then locks ADCH
    uint8_t high = ADCH; // unlocks both
```

```
long result = (high<<8) | low;

result = 1125300L / result; // Calculate Vcc (in mV); 1125300 = 1.1*1023*1000
return result; // Vcc in millivolts
}
```

```
void beep(int note, int duration)
{
    //Play tone on buzzerPin
    tone(9, note, duration);
```

```
    delay(duration);
```

```
    noTone(9);
```

```
    delay(50);
```

```
}
```