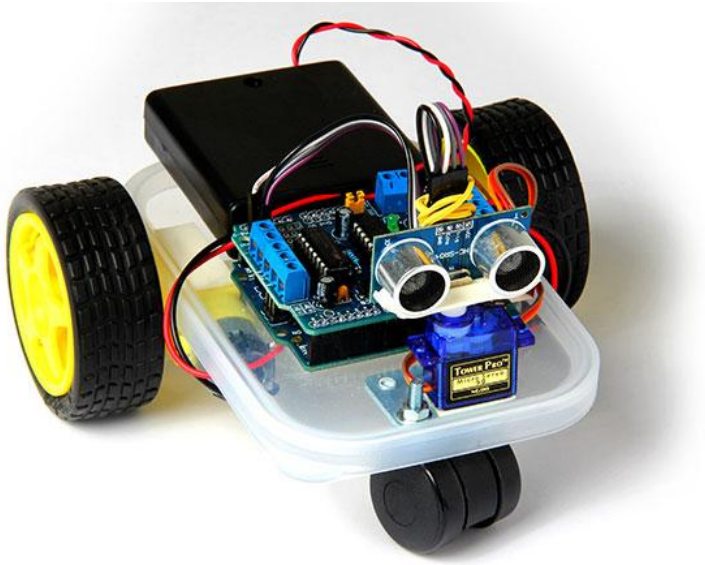
EP486 Microcontroller Applications

Topic 8

Using Arduino Data in C++ and in Processing



Department of Engineering Physics

University of Gaziantep

Nov 2013

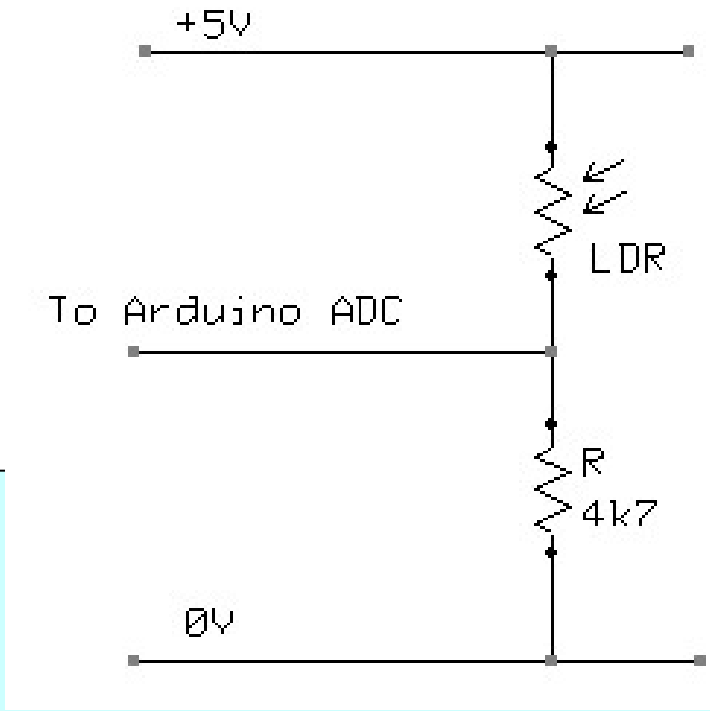
Introduction

Here we'll see some applications of basic DAQ in

- Processing
- Standalone C++

The data is generated by Arduino via LDR resistance level.

Arduino LDR (code)




```
int sensorValue = 0;

void setup(){
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  sensorValue = analogRead(A0);

  Serial.println(sensorValue);
  delay(100);
}
```

Processing (code)

```
// Example code for Arduino --> Processing
import processing.serial.*;
Serial myPort;  // The serial port
S2I    myConv;  // The Str to Int Converter (see web page)
String inString ="";

void setup() {
   println(Serial.list());
   myPort = new Serial(this, Serial.list()[1], 9600);
   myConv = new S2I();
   size(300,300);
}

void draw(){
   if (myPort.available() > 0) {
    inString = myPort.readString();
    int val  = myConv.StrToInt(inString);
    println(val);
    background(4*val, 0, 0);
  }
  delay(100);
}
```

C++ (code)

```
// Example code for Arduino --> Standalone c++
#include <iostream>
#include <string>
#include <tchar.h>
#include "SerialClass.h" // Library (see course web page)
#include "SerialClass.cpp"
using namespace std;

// application reads from the specified serial port
int _tmain(int argc, _TCHAR* argv[]){
        cout << "Welcome to the serial test app!" << endl;
        Serial* SP = new Serial("\\\\.\\COM6"); // adjust as needed
        if (SP->IsConnected()) cout << "We're connected" << endl;

        const int dataLength = 100;
        char incomingData[dataLength] = "";
        int readResult = 0;
        while(SP->IsConnected())
        {
            readResult = SP->ReadData(incomingData, dataLength);
            int  val   = SP->StrToInt(incomingData, dataLength);
            cout << val << endl;
            Sleep(100);
       }
}
```