```
#include <wiringPi.h>
#include <iostream>
#include <unistd.h>
#include <sys/select.h>
#include <sys/time.h>
#include <string>
#include <cstdlib>
#include "mcp3008Reading.h"
//#include "archives.h"

using namespace std;

int main(int argc, char* argv[]){
int cycle = 3600;

if(argc == 2){
        cycle = atoi(argv[1]);
}

//string filename("/home/pi/Documents/moisture_sensor_TML/mcp.dat");
int tab[3];

//Setup WiringPi in order to use the library wiringPi

wiringPiSetup();
//pinMode(0,OUTPUT);    //
pinMode(2,OUTPUT);      //Relay control

const int moistureChannel = 5;
const int moistureLimit = 500;


//Initialisation of the Virtual Pins
initMCP3008();

while (1){
        int x = readMCP3008(moistureChannel);
        //double temper = analogRead(BASE+6);
        //cout << "temper = " << temper << endl;
        //temper = temper / 1024 * 3.3 * 10;

        struct timeval tod;
        gettimeofday(&tod,NULL);
        tab[0] = tod.tv_sec;

        if (x < moistureLimit){ // The soil is moist enough
        digitalWrite (2,LOW);
        }
        else{
        digitalWrite (2,HIGH);  //We activate the pump
        }

        tab[1] = x;
        tab[2] = 0;
//      tab[2] = (int)temper;
        cout << "time : " << tab[0] << ", Value is: " << tab[1] << ", and TÂ° is : " << tab[2
]  << " ÂºC"<< endl;

//      saveInFile(filename, tab,(unsigned int)( sizeof(tab)/sizeof(*tab) ));

        struct timeval t;
        t.tv_sec = cycle;
```



```
        t.tv_usec=0;
        select(0,NULL,NULL,NULL,&t);

}
cout << "returning 0 " << endl;
return 0;

}
```

```
#include <wiringPi.h>
#include <mcp3004.h>
#include "mcp3008Reading.h"
using namespace std;

#define BASE 100
#define SPI_CHAN 1

int initMCP3008(){
        return mcp3004Setup(BASE,SPI_CHAN);
}

int readMCP3008(int channel){
//      wiringPiSetup();
//      mcp3004Setup(BASE, SPI_CHAN) ;
        int value = analogRead(BASE+channel);
        return value;
}
```

```
#ifndef MCP3008READING_H
#define MCP3008READING_H

#define BASE 100
#define SPI_CHAN 1

int initMCP3008();

int readMCP3008(int channel);

#endif
```

```
CXXFLAGS=-Wall #Define CXXFLAGS to automatically add them in the command

SRC=$(wildcard *.cc)
DEPS=$(wildcard *.h)
LDFLAGS=-lwiringPi
OBJ=$(SRC:.cc=.o)

# Redefine the default command to create executable without suffix (use g++ inst
ead of gcc)
awps: $(OBJ)
        g++ $(LDFLAGS) -o $@ $^

all: awps awps.pdf

awps.pdf: $(SRC) $(DEPS) Makefile
        a2ps -o - $^ | ps2pdf - docs/$@

clean:
        rm -f *~ *.o *.bak

mrproper: clean
        rm -f awps

depend:
        makedepend $(sources)
```