Name : chima wisdom enyiche

Matric number : 18/eng04/025

Depertment: electrical /electronics engineering

Course: eng 224

Question 1: discuss the application development following the softwear development cycle

The software development cycle include :

- Conceptualization
- Specification
- Design
- Implementation
- Testing and debugging

- Release and update

Conceptualization: the application to be developed would work simultaneously _ with the irrigation machine. It is an automated system which require user input .the security system would be a number password system developed with visual basic ,this password system would link to the backend program which would be coded in c language .the application would be linked with different sensors which all collect data which would serve as input to the program .these sensors would include a temperature sensor, a moisture sensor ,a watering system,a pressure sensor, and an LED . the temperature and moisture sensors would collect data at intervals which would help the application determine the actions of the watering system .the pressure sensors would would be placed at the base of the water tank .the volume of water in the tank at full capacity would be 0.86m^3 and the pressure would be 10.747n/m^2 . so using boyld's law (p1*v1=p2*v2) where p2 is the pressure being recorded by the pressure sensor on the tank , the volume of the water in the tank can be calculated by the app and if this volume falls bellow 25% of the original volume (at full capacity) an LED would be turned on to indicate the water level being low .

Specification :

The specification must break the application down into moduls.

Hardwear components:

- Temperature sensor: the temperature sensor would take the temperature of the soil , this value would be updated every second in order to account for little variations.the sensor would take the temperature in an analogue form and a analogue to digital converter would be used to change it toa form the computer understands . The Temperature Sensor used is LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature.
- Soil moisture sensor: a Soil moisture sensor FC-28. This sensor measures the volumetric content of water inside the soil and gives us the moisture

level as output. The sensor is equipped with both analog and digital output, so it can be used in both analog and digital mode.

- Pressure sensor: this pressure sensor pm-33 would be used to measure the pressure exerted by the water on the bottom of the tank. This data would be collected by the code and used to calculate the volume of water in the tank which would be displayed on the system gui
- Watering system: this is the key part of the hard wear systems .it reacts to the data collected and is on a repeted timer based on the last use
- LED:the led would serve as the alarm system ,indicating a low level o water .it comes on when the volume of water in the take is lover that 25% the original value

soft wear components:

- GUI: the graphical user interface would be created with visual basics and would consist of two forms ,one housing the password system and the other the data base system .
- Programing language: the language used are visual basic and c language . the C-coded backend logic is made to a DLL and an executable VisualBasic is created . the latter is let to call the DLL based C functions from the VisualBasic exe.the c program is linked to each individual sensor and specific data is extracted

Design

In the design stage various conditions are tested , an alternative path of procedure is chosen based on the outcome of the tests and the sequence of these tests are determined by two ways , which are (1) algorithm (2) flowchart

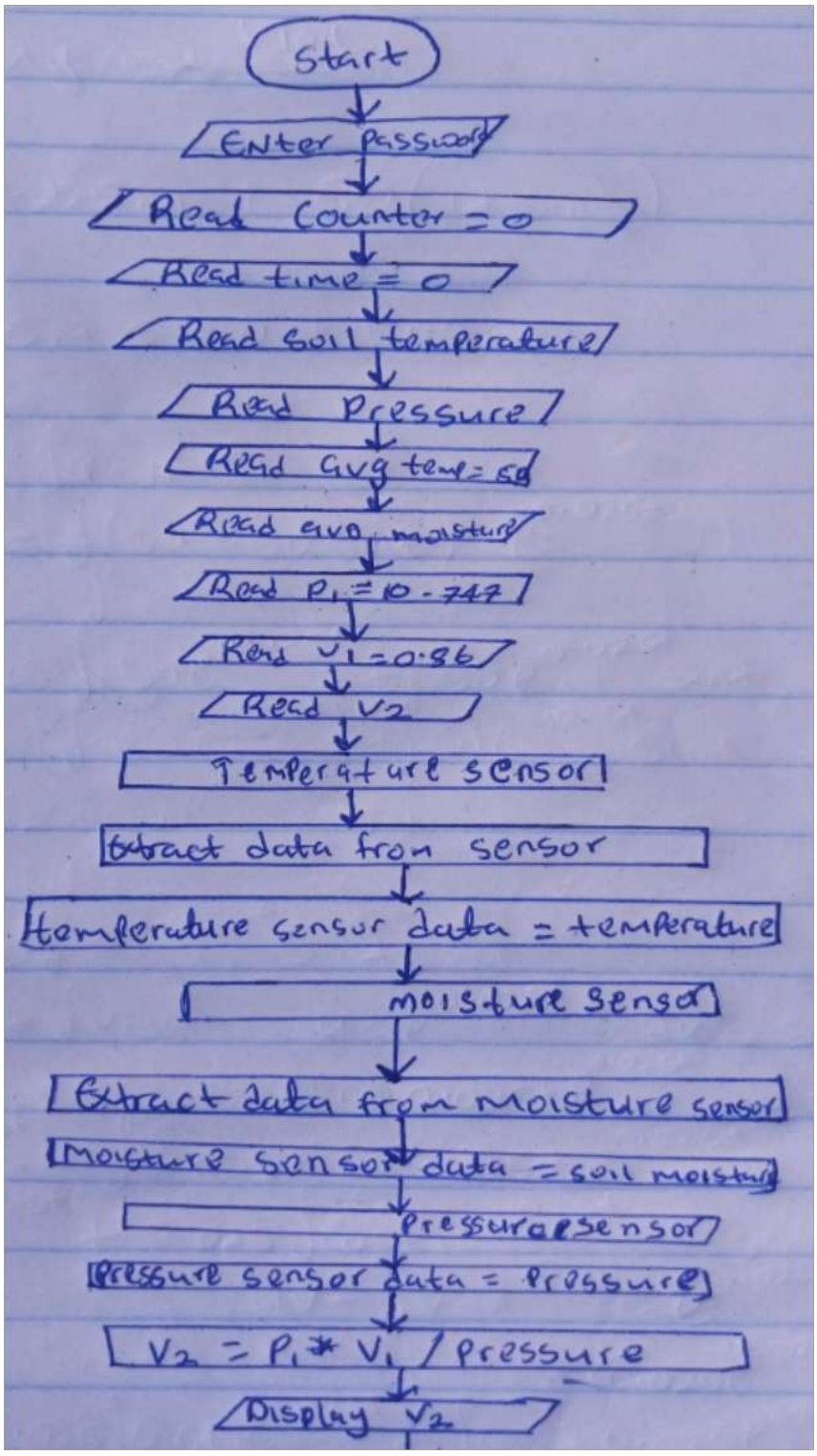
Algorithm :

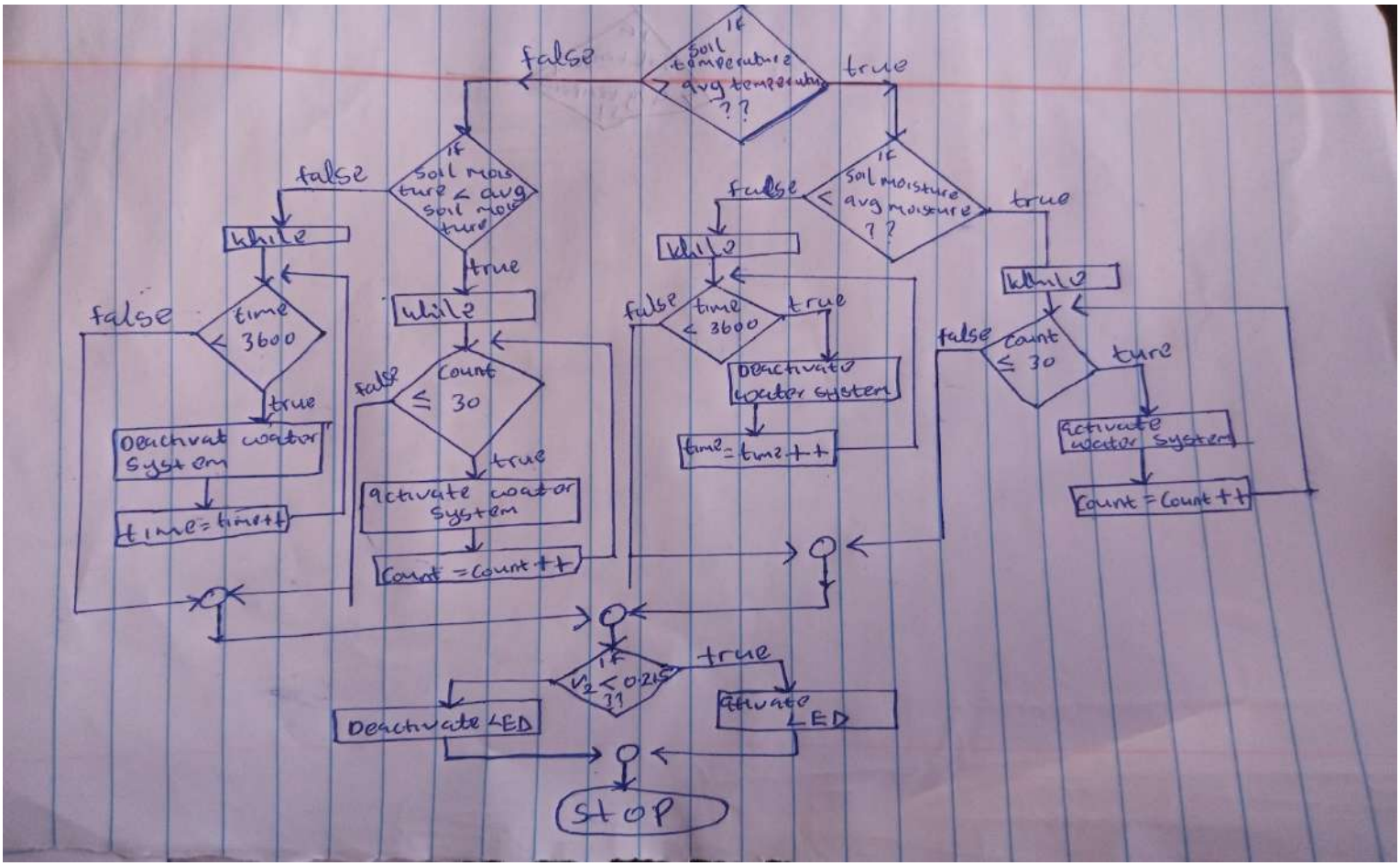
1. Start
2. Enter passcode
3. Read count=0
4. Read time=0

5. Read soil temperature
6. Read pressure
7. Read avg temp= 50
8. Read avg moisture=50
9. Read soil moisture
10. Read p1=10.747
11. Read v1=0.86
12. Read v2
13. Temperature sensor
14. Extract data from sensor
15. Extracted temperature data = temperature
16. Moisture sensor
17. Extract data from moisture sensor
18. Extracted moisture data =soil moisture
19. Pressure sensor
20. Extract data from pressure sensor
21. Extracted pressure data = pressure
22. V2=p1*v1/pressure
23. Display v2
24. If soil temperature > avg temperature then
25. If soil moisture < avg moisture then
26. While count <= 30
27. Activate watering system
28. Count = count ++

29. Else
30. While time < 3600
31. Deactivate watering system
32. Time = time++
33. else
34. If soil moisture < avg moisture then
35. While count <= 30
36. Activate watering system
37. Count = count ++
38. Else
39. While time < 3600
40. Deactivate watering system
41. Time = time++
42. If v2 < 0.215 then
43. Activate LED
44. Else
45. Deactivate LED
46. Stop

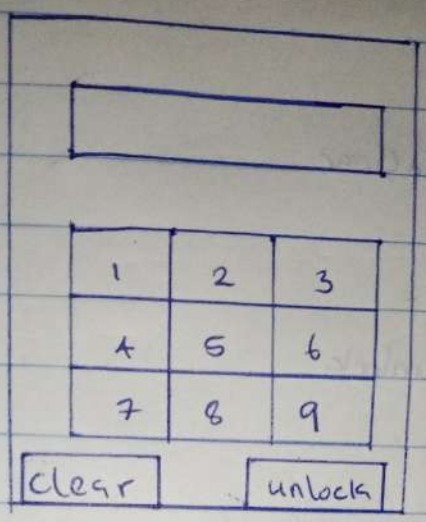
Flowchart


Start
Enter password
Read Counter = 0
Read time = 0
Read soil temperature
Read Pressure
Read avg temp = 58
Read avg moisture
Read P1 = 10.747
Read V1 = 0.96
Read V2
Temperature sensor
Extract data from sensor
Temperature sensor data = temperature
moisture sensor
Extract data from moisture sensor
Moisture sensor data = soil moisture
Pressure sensor
Pressure sensor data = Pressure
V2 = P1 * V1 / Pressure
Display V2



Implementations

Visual basic CODE



Caption Properties

→ Label (Password)
Object name = lblPass
Caption = " "

→ Command (1)
Object name = Cmd1
Caption = 1

→ Command (2)
Object name = Cmd2
Caption = 2

→ Command (3)
Object name = Cmd3
Caption = 3

→ Command (4)
Object name = Cmd4
Caption = 4

→ Command (5)
Object name = Cmd5
Caption = 5

→ Command (6)
Object name = Cmd6
Caption = 6

→ Command (7)
Object name = Cmd7
Caption = 7

→ Command (8)
Object name = Cmd8
Caption = 8

→ Command (9)
Object name = Cmd9
Caption = 9

→ Command(Clear)
Object name = CmdClear
Caption = Clear

→ Command (Unlock)
Object name = Cmd unlock
Caption = unlock

Code

```
→ Private Sub Cmd1 - Click()
  lblPass.text += "1"
  End Sub
→ Private Sub Cmd2 - Click()
  lblPass.text += "2"
  End Sub
→ Private Sub Cmd3 - Click()
  lblPass.text += "3"
  End Sub
→ Private Sub Cmd4 - Click()
  lblPass.text += "4"
  End Sub
→ Private Sub Cmd5 - Click()
  lblPass.text += "5"
  End Sub
→ Private Sub Cmd6 - Click()
  lblPass.text += "6"
  End Sub
```

```
→ Private Sub Cmd7 - Click()
  lblPass.text += "7"
  End Sub
→ Private Sub Cmd8 - Click()
  lblPass.text += "8"
  End Sub
→ Private Sub Cmd9 - Click()
  lblPass.text += "9"
  End Sub
→ Private Sub CmdClear - Click()
  lblPass.text = ""
  End Sub
→ Private Sub Cmdunlock - Click()
  Dim Password as String
  Password = "1111"
  If lblPass.text = "1111"
    hw.dll.run
  Else
  lblPass.text = ""
  End Sub
```

C programing code:

```c
#include <stdio.h>
#include <stdbool.h>
#include <serf.h>
float temp;
int tempPin = 0;
void setup()
{
   Serial.begin(9600);
}

void loop() {
   temp = analogRead(tempPin);
   // read analog volt from sensor and save to
     variable temp
   temp = temp * 0.48828125;
   Serial.print("TEMPERATURE = ");
   Serial.print(temp);
   Serial.print("*C");
   scanf(temp)
   Serial.println();
   delay(1000); // update sensor reading each one
     second
}
int sensor_pin = A0;
int output_value ;
void setup() {
  Serial.begin(9600);
   Serial.println("Reading From the Sensor ");
  delay(2000);
   }
void loop() {
   output_value= analogRead(sensor_pin);
   output_value = map(output_value,550,0,0,100);
   Serial.print("Mositure : ");
   Serial.print(output_value);
   Serial.println("%");
   scanf("%")
   delay(1000);
   }
   const float  OffSet = 0.483 ;

float V, P;

void setup()
{
  Serial.begin(9600);
  Serial.println("Water pressure sensor ");
}
void loop()
{
  V = analogRead(0) * 5.00 / 1024;     //Sensor
    output voltage
  P = (V - OffSet) * 400;             //Calculate
    water pressure
  Serial.print("Voltage:");
  Serial.print(V, 3);
  Serial.println("V");
  Serial.print(" Pressure:");
  Serial.print(P, 1);
  Serial.println(" n/m^2");
  scanf(p,1);
  Serial.println();
  delay(500);
}
int motorPin = A0; // pin that turns on the motor
int watertime = 5; // how long to water in seconds
int waittime = 60; // how long to wait between
  watering, in minutes
void setup()
{
    pinMode(motorPin, OUTPUT); // set A0 to an
      output so we can use it to turn on the
      transistor
}
{
    int led = 13;
    int avg temp = 50;
    int avg mosit = 50;
    double  p1=10.747
```

```
73      double  v1=0.86
74      double v2=(p1 * v1)/p
75      double v low = 0.215
76      void loop()
77      {
78      temp > avg temp ? % < avg moist ? digitalWrite(
          motorPin, HIGH) : digitalWrite(motorPin, LOW)
           : % < avg moist ? digitalWrite(motorPin,
          HIGH)  :digitalWrite(motorPin, LOW);
79          }
80          if(digitalWrite(motorPin, HIGH))
81          {
82              delay(watertime*1000)
83          }else
84          {
85              delay(waittime*60000)
86              }
87              void setup()
88    {
89          pinMode(led, OUTPUT); // Set pin 13 as
            digital out
90     Serial.begin(9600);
91       Serial.flush();
92  }
93  void loop()
94      {
95          if (v2<v low)
96             digitalWrite(led, HIGH); // on
97          }else
98          {
99             digitalWrite(led, low);
00       }
```

Testing and debugging:

A computer program may not run properly the first time due to the

possibility of logical and/or syntactic errors. These errors are referred as bugs. The process of removing these bugs is called debugging. The errors in the program are checked by testing it at various stages.

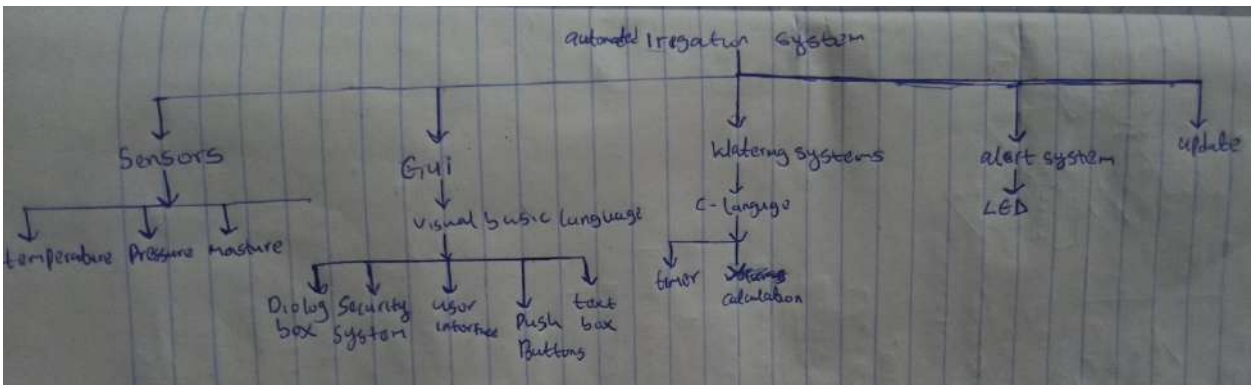
o Use of Test Data: Test data should be prepared in such a way that the programs run by using this data. It should be prepared so as to include all the conditions that the program is expected to test.

o Use of Diagnostic Tools: There are diagnostic aids, which programmers may deploy to detect errors when a program fails to run correctly. The diagnostic procedures may vary from one language to another. The diagnostic tools provide, in general, a method for testing the execution of a program at each step or each time the program follows a particular path. Breakpoints may also be used in code for debugging purposes.

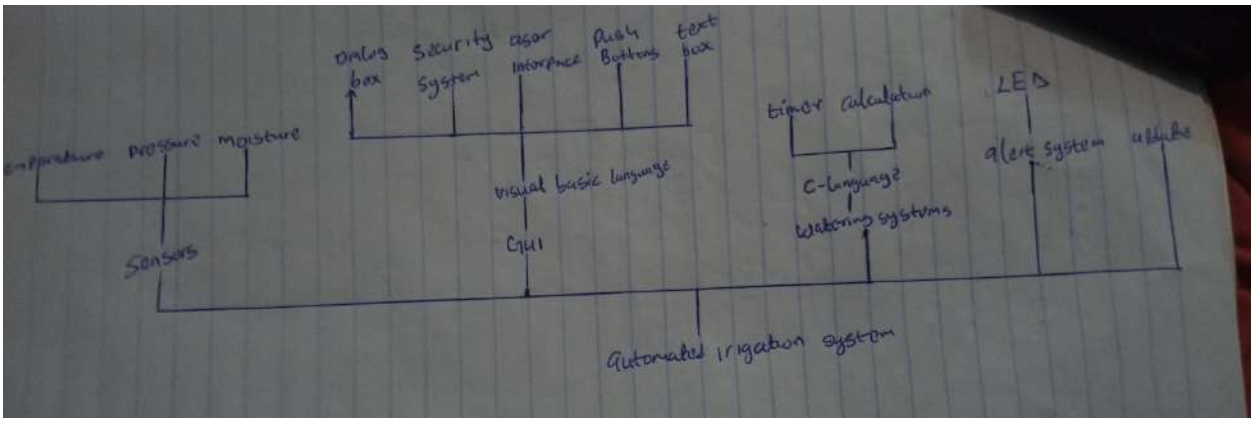
Release and update

The soft wear is released for used and tested with real time data and update based on user feed back

Top-down design



Bottom -up design


Dialog box
Security system
User interface
Push Buttons
text box
timer calculation
LED
temperature
Pressure
moisture
visual basic language
C-language
alert system
Sensors
GUI
Watering systems
Automated irrigation system