KOCAELİ UNIVERISTY*FACULTY OF ENGINEERING

SmartFarm App

Engineering Design -3 Project Report

Metehan Karahalilöz
180207028

Dept: Electronics and Communication Engineering

S u p e r v i s o r : Assoc. Prof. Dr. Oğuzhan Karahan

KOCAELI, 2023

TABLE OF CONTENTS

1. Introduction

Good luck on the human way of keeping up with technology and technology in today's world. In our project, we are working in settlements such as villages and towns

It is a project to use it for human treatment in their lives, thinking about our farmers who live and make a living through agriculture and animal husbandry. Smart a g r i c u l t u r e , agriculture It is a management concept that focuses on infrastructure to l e v e r a g e advanced technology, including big data, cloud and the internet of vision (IoT) to monitor, track, automate and analyze center operations. precision agriculture Smart agriculture, also known as smart f a r m i n g , is managed by software and monitored by sensors. the importance o f smart farming is increasing due to the growing global population, the increasing demand for higher yields, the need to use natural productive resources, the increasing use and complexity of information and communication technology, and the increasing climate smart phone applications.

1.1. Smart farm technologies

A smart farm includes technologies such as Land uses and water, l i g h t ,

Sensors for humidity and temperature management. Advanced networking and telecommunication technologies such as GPS.

Achieving specialized applications and IoT-based solutions, robotics and automation

Data analytics tools for decision making and forecasting. Crop

Data collection is an important part of smart farming as the amount of data from cropping, mapping, climate change, fertilizer soil applications, weather data, machinery and animal health continues to grow. Satellites and drones to collect round-the-clock data for an entire field. This information is transmitted to IT systems for monitoring and analysis to give an "eye in the field" or "eye in the store", making remote monitoring possible.

The combination of these technologies f a c i l i t a t e s machine-to-machine (M2M) derived data. From this source, it feeds into a decision support system, so that farmers have a more granular view of what is going on than in the past. For example, in a field
By precisely measuring needs and adapting strategy accordingly, farmers
The use of pesticides and fertilizers is greatly improved and can be used more similarly.
Similarly, smart farming parenting allows farmers to keep individual animals
It helps them to better monitor their traits and adjust their nutrition for their herd flocks, including restrictions for diseases.

1.2. Benefits of a smart farm

Precision agriculture, making farming more connected and smarter, overall costs
It helps to reduce and conserve and consume, to ensure the sustainability of agriculture and the user experience. Increased control over production, better cost management and flowers, leading to a reduction in quantity. Controlling the monitoring of abnormalities, for example in crop organs or animal health, helps to eliminate the risk of data loss. In addition, i t increases the possibilities for automation. With smart devices, multiple processes can be activated at the same time, and automated services allow for better control of production.
by increasing product quality and volume. Smart farming systems also increase demand
Careful evaluations of the forecast and assessments to manage wastage ensure timely market placement. Precision farming, pay attention to managing land supply to achieve production for the right crop in demand and focus on the right growth, such as moisture, fertilizer or material content, depending on the ways the land is farmed.
The types of sensitive agricultural products applied, the software for consumption management depends on t h e use. Control systems manage sensor input, providing remote information for supply and decision support, in addition to automating machinery and equipment for emergent r e s p o n s e and production support.

2. PLATFORM USED

2.1. Ingredients

In order to build a smart farm we need to have read data, we need to transmit our data to sensors. We also need to have certain documentation of the systems we are going to build to be able to respond to this capability.

2.1.1. ESP 32

ESP32 can operate as a completely independent system or as a device connected to a host MCU, reducing the communication load on the host application processor.
ESP32 is executed with other systems to obtain SPI/SDIO or I2C/UART outputs via Wi-Fi and Bluetooth.

Figure 1 ESP32

2.1.2. DHT11

The DHT11 is a basic, ultra-low temperature digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to sense the surrounding air and outputs a digital signal to the data pin (no analog input pins are needed). It is quite long to use, but requires careful timing to get the data.

Figure 2 DHT11

2.1.3. MQ-2

MQ-2, a smoke and combustible gas sensor from Winsen. 300 - 10000ppm combustion range can detect gases. The most common use is for domestic gas leak alarms and detectors with high sensitivity to propane and smoke.

Figure 3 MQ-2

2.1.4. LDR

Photoresistors, also known as light dependent resistors (LDRs), are generally used to resist the incident light or are storage-sensitive devices that allow you to monitor their absence or consume their waste.

Figure 4 LDR

2.1.5. SERVO MOTOR

Servo motors, or "servos" as they are known, are a machine tool that allows a machine are electronic devices and rotary or steering actuators that rotate and propel in a rotary or steering manner. Servos a r e mainly used for angular or trajectory tracking and for specific speed and acceleration.

Figure 5 SERVO MOTOR

2.1.6. DC MOTOR

A DC motor is any class of rotary electric motor that converts current-directed (DC) electrical energy into mechanical energy.

Figure 6 DC MOTOR

2.2. Written m

Software i s a set of instructions, data or programs that you use to operate computers and use certain limits. A computer can be hacked using
is the opposite. S o f t w a r e is a general term for having applications, scripts and programs running on a device.

2.2.1. C#

C# (pronounced "See Sharp") is a modern, objectively behaving and type-safe programming language. C# enables developers to build . NETWORK. C# has its roots in the C language family and will be immediately familiar to C, C++, Java and JavaScript programmers.

During the compilation of the .NET Framework, the class libraries were originally compiled using a managed code compiler system called "Simple Managed C" (SMC). [20][21] In January 1999, Anders Hejlsberg formed a team to create a new language called Cool, which at the time stood for "C-like Object Oriented Language."[22] Microsoft had considered keeping "Cool" as the name of the language's son, but chose not to for trademark reasons. .NET project July 2000 Professional

When it was publicly announced at the Developers Conference, the language was renamed C#. It was requested and the class libraries and ASP.NET runtime w e r e ported to C # .

- Transportability

By design, C# is the programming language that most directly controls the underlying Common Language Infrastructure (CLI).[62] Intrinsic organization plural, the value applied by the CLI framework

types. However, the language specification does not require the compiler to generate code. does not specify data transfers: that is, whether a C# compiler generates a Common Language Runtime Target or a Common Intermediate Language (CIL) or any other special format.

should generate machine code. Theoretically, a C# compiler can generate machine code just like traditional C++ or Fortran compilers.

- Text m

The C# language does not allow global variables or implementation. All invitations and member classes must be notified within Static accessories of general classes may replace general equipment and accessories.

Unlike C and C++, local variables cannot shadow partition variables.

- Methods and functions

The C# language is a method, a member of a class whose parts can be implemented as a function (a sequence of parts) from the mere use of a class property to hold a value. Like C++ and ANSI C

As in other syntactically similar languages, the signature of a method is a declaration that contains the following any runtime-dependent accessibility keyword (such as private), the e x p l i c i t specification of the translation (such as int or keyword invalid), the name of the method, and the last as, depending on the type, official records and pendants of each one parameter, none of them

An array within parentheses of c o m m a - s e p a r a t e d parameter
specifications, consisting of a global value, when not present. Issued. Certain specific
types, such as those that take or set using a class property by return value or
assignment
r e a d i n e s s , full signature documents, but in the general case, the definition of a class
includes a full signature declaration of methods.

• Feature

C# has been modified to have extension classes. Features can be implemented in a support area,
having a simple accessor tap or receiver and set items.
Since C# 3.0, the syntactic structure of automatically applied properties, where t h e accessor
(receiver) and modifier (setter) operations encapsulate operations on a single attribute of a class, is
syntactically
sugar is available.

• Name field

A C# namespace i s a Java package or a C++ package, with rules and properties very similar to a package.
provides the same level of code warmth as a namespace. Ad spaces can be internalized with the
syntax "using".

• Memory

In C#, memory address pointers can o n l y b e u s e d f o r blocks specifically marked as pills.
[69] and programs containing non-safe code require appropriate permissions to run. Most
object accesses are always either pointing to a "live" object or to a well
safe object references with null identifier value; "dead"
cannot obtain a reference to an object (a garbage-collected object) or to an arbitrary memory
register. An unsafe pointer is a pointer to a garbage-collected object, array, array or
Any reference to an organization-allocated memory library may indicate instantiation when the centers
return an 'unmanaged' value. Code that is not marked as insecure, type System.IntPtr
can store and process beacons, but c a n n o t de-operate them.

• Exceptions

A number of standard exceptions are available to programmers. Guidance in standard
libraries, some builds throw system exceptions on a regular basis, and the range o f
exceptions thrown is documented incidentally. Special exception class guides and sections
for classes
processing will be allowed for specific cases.

Checked exceptions are not available in C# (unlike Java). This was a board decision based on crawlability and versionability considerations.

- architecture

Unlike C++, C# does not support multiple inheritance, although it can implement multiple "interfaces" (completely abstract classes) in a class list. This was done by the chief architect of the language to simplify the architectural requirements throughout the CLI to avoid constraints and overhead.

was a decision on a bill.

2.2.2. C# FORM APPLICATION

A Windows formula in C# is a form that runs on a computer.

Visual Studio Form with C# can be used to create a Windows Forms application.

Controls can be added to Windows C# forms via the Toolbox tool in Visual Studio.

Controller such as labels, checkboxes, radio button, etc.

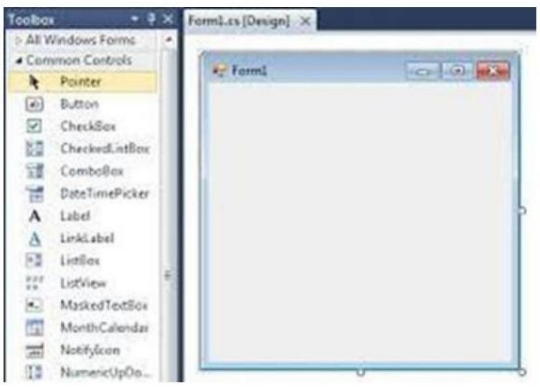

Figure 7 C# FORM EXAMPLE

3. IoT (Internet of Things)

The Internet of Things, or IoT as it is called in English, is a communication network that connects physical components or larger systems.
Connectivity to other devices and systems and objects over the Internet
embedded with sensors, s o f t w a r e and other technologies to exchange data. It is envisioned that objects can be marked with a single key and w o r k together via internet controllers, so that t h e sum of small results can add up to larger values. For the military system, the "military intelligence internet"
detached
Traditional embedded systems, wireless sensor networks, control systems, automated (including home automation and building automation) and other areas are communicated by enabling the Internet of Things.
In terms of the Internet of Things, the concept of object has a very broad meaning. Any monitoring device, sensor, biochip or access device qualifies as an object. A device
The necessary condition for it to be considered "smart" and qualify as an object: it must have a birth name (unique id), be connectable and have a sensor.
This makes the smart object accessible and controllable from anywhere in the world. Due to the limited number of commonly used IP addresses
It is not possible to have a fictitious unique name, and by using IPv6 designed to overcome this, the application can become truly unique.
IoT applications are used to make sensors accessible individually, as well as to extract value by combining many sensor arrays. After evaluations of the large amount of sensor data flowing through physical environments (data)
the transmission of information to operators or relevant users as information, or the u s e o f data systems to provide an activity by processing.
The first devices that brought children into the global network appeared in 1982. It was a Coca-Cola vending machine that could control the machine's operation and keep track of what it could dispense. Kevin Ashton
by an RFID technology researcher in 1999, the invitation to the "Internet of Things" was formulated by the Seven Sevens.

content related to concepts, d i s c u s s i o n s and individual concepts. The 2000s and 2010s were a period of rapid development, when IoT projects started to be successful and some practical applications were rapidly developed. Many small and large projects were built, from smart lights and sports trackers to self-driving cars and s m a r t cities. This was also a period of rapid development with the emergence of wireless connections capable of transmitting information over long distances and the increasing bandwidth of Internet communications. was possible because of its breadth.

IoT has evolved into a completely "different internet" in which all existing protocols benefit and connectivity. Therefore, it has become a vital necessity to create protocols and standards for restricted IoT communication. However, some existing technologies (e.g. HTTP) are also used by the Internet of Things.

3.1. Serial Communication

In serial communication, the sections are in the direction of binary pulses. In other words, binary 1's l o g i c can be high or 5 volts, 0's logic can be low or 0 volts. serial communication,

Acquisition can take various forms depending on what they are receiving and the type of data acquisition. This acquisition mode i s spread across unidirectional, semi-bidirectional and full bidirectional. For each learning mode there is a

there i s a transmitter and receiver. There i s a transmitter and receiver for each receive mode.

Figure 8 Serial Communication

4. System Installation

The installed system reads our data from the UI together with the electronics as well as the system data at the same time.

4.1. Conclusion

View our interface in Figure 9.

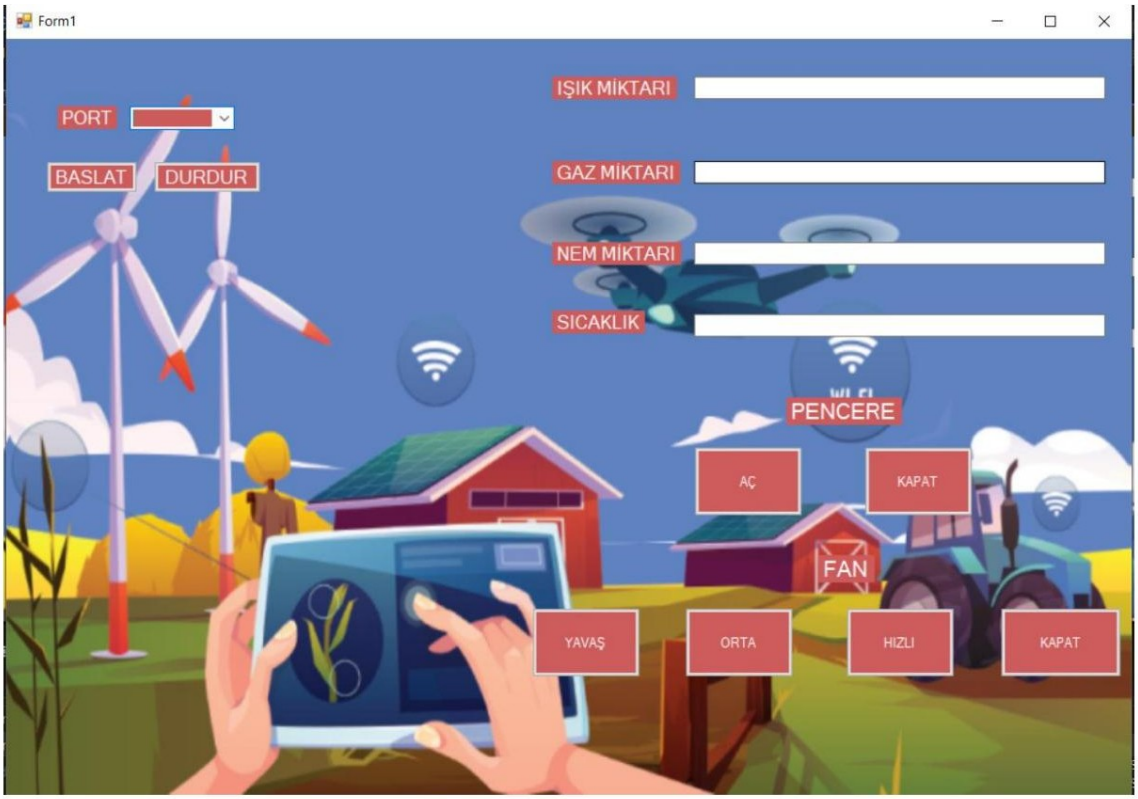

Figure 9 user interface

Figure 10 contains the libraries necessary to communicate with the form and to r u n our form properly.

```
1   using System;
2   using System.Collections.Generic;
3   using System.ComponentModel;
4   using System.Data;
5   using System.Drawing;
6   using System.IO.Ports;
7   using System.Linq;
8   using System.Reflection.Emit;
9   using System.Text;
10  using System.Threading.Tasks;
11  using System.Windows.Forms;
```

Figure 10 Library

The codes required for system and computer communication are written in Figure 11.

```
private string data;

public Form1()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    string[] ports = SerialPort.GetPortNames(); //Port isimlerini al
    foreach (string port in ports)
        comboBox1.Items.Add(port); //Port isimlerini comboBox içine yaz
    serialPort1.DataReceived += new SerialDataReceivedEventHandler(SerialPort1_DataReceived);
}

private void SerialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    data = serialPort1.ReadLine();                    //Veriyi al
    this.Invoke(new EventHandler(displayData_event));
}
```

Figure 11 Serial Communication

Figure 12 shows the codes that read the data in the system and the comparison of these values with the values that should be there.

```
private void displayData_event(object sender, EventArgs e)
{
    string[] value = data.Split('*'); //'*' gördüğün yerlerden stringi ayır ve diziye ata
    textBox1.Text = DateTime.Now.ToString()+ "  " + "IŞIK DEĞERİ " + value[0];
    textBox2.Text = DateTime.Now.ToString()+ "  " + "GAZ  DEĞERİ " + value[1];
    textBox3.Text = DateTime.Now.ToString()+ "  " + "NEM  DEĞERİ % " + value[2];
    textBox4.Text = DateTime.Now.ToString()+ "  " + "SICAKLIK DEĞERİ " + value[3] + "°C";
    int a = 4095;
    int ısık = Int32.Parse(value[0]);
    int b = 650;
    int gaz = int.Parse(value[1]);
    int c = 50;
    int nem = int.Parse(value[2]);
    int d = 20;
    int sıcaklık = int.Parse(value[3]);
    if (ısık >= a)
    {label10.Text = "IŞIĞI AÇ";}
    else
    {label10.Text = string.Empty;}
    if (gaz > b)
    {label11.Text = "CAMI AC";}
    else
    { label11.Text = string.Empty;}
    if (nem > c)
    {label12.Text = "FANI ÇALIŞTIR";}
    else
    { label12.Text = string.Empty;}
    if ( sıcaklık > d)
    {label13.Text = "FANI ÇALIŞTIR";}
    else
    {label13.Text = string.Empty;}


}
```

Figure 12 Data retrieval

Bibliography

[1] Güneş, H., Bı çakcı , S., (2018). Voice Command Perception Methods for Fluid Houses. Journal of Balı kesir University Graduate School of Science and Technology, 20(2), 561- 568.

[2] Jamalabad, MS, (2014). Akı Houses Producing Energy from Climate. Master's Thesis, Supervisor Işı k, B., Aydı n University, Istanbul, 8-21.

[3] Yumurtacı , M., Keçebaş, A., (2009). Flux Home Technologies and Automation Systems Flux Building Technologies and Automation Systems, 5th International Advanced Technologies Symposium (IATS'09), Karabük, 1-6.

[4] Melek T. ,(2019). Innovative Design i n Housing: An Evaluation on Akı Houses. Master's Thesis, Supervisor Prof. Dr. Neslihan D., Istanbul Kültür University, Istanbul, 26-192.