Portable Farming

Sutono\textsuperscript{1*}, Selvia Lorena BR Ginting\textsuperscript{2}
\textsuperscript{1,2}Departemen Teknik Komputer, Universitas Komputer Indonesia, Indonesia

Email: *sutono@email.unikom.ac.id

Abstract. The purpose of the design of Portable Farming is to utilize home yards, especially in urban areas where land is limited, the concept of Portable Farming is essentially adopting Smart Farming technology which is currently booming. By combining Smart Farming technology and Portable Land, the idea of designing Portable Farming was created. Portable land is a container that is designed to be an agricultural land with a certain volume so that it can be moved to another location without the need to change the farming system. Portable Farming is my own research that has not been proven true whether the merging of the two Smart Farming technologies and Portable Land can be used as a solution to create green land, especially in urban areas with limited land.

1. Introduction

One of the advantages of Smart Farming is the fast vegetable harvest time. Conventional agriculture usually takes about 40 to 60 days to reach the harvest period. It also still depends on the weather and the possibility of pests and diseases. Whereas Smart Farming only takes about 18 days to harvest small vegetable varieties and 33 – 35 days for clean and healthy large vegetable varieties. Because one of the factors needed for plant growth besides nutrition and water is sunlight. Smart Farming uses a system that can regulate temperature, water source, humidity, nutrition and special Growlight type LEDs that replace sunlight [1-6]. This system is considered to save water usage. The light coming from this special LED can make it easier for plants to photosynthesize because it lights up at home temperature. The process of converting light energy into chemical energy for plant needs, this method is beneficial to challenging environments such as deserts, mountains or large cities that require more money for labor costs. Smart farming can produce 3,000 to 3,500 tons of vegetables each year.

Portable Land is a container or a place with a certain volume that is used for agricultural land, this container is made of acrylic material so that it does not pose a danger of corrosion to agricultural land. The design of this container is shaped like an open car body that has four wheels like a car so that it can be moved to a particular location by pulling or pushing the container to a certain location, therefore this land is called a protected land [7].

By combining the concept of Smart Farming technology and Portable Land, a system known as Portable Farming has been designed that has been equipped with a soil moisture sensor that is integrated with the IoT device, so we can monitor the condition of the farm anytime and anywhere we are using the application BLYNK can be downloaded at Playstore.
2. Method
Because this research is a research that has never been done by researchers before, we only tried to combine from existing technology, namely Smart Farming and Portable Land into a new research concept that we call Portable Farming. So in this discussion we only discuss the problem of device requirements and how to build the Portable Farming system.

Portable Farming is a certain volume of agricultural land that is designed using IoT technology and Arduino Mega 2560 with Soil Moisture Sensor FC-28 that is able to detect soil moisture. The Soil Moisture Sensor needed in this system is 12 pieces with a plant area of about 30 cm² each that forms a 3x4 matrix or requires an area of 108 m². Soil Moisture Sensor FC-28 has an input voltage specification of 3.3 volts or 5 volts, an output voltage of 0 - 4.2 volts with a current of 35mA and an ADC value range of 1024 bits ranging from 0 - 1023 bits [8-12] (see Figure 1).

![Soil Mesture Sensor FC-28](image1)

**Figure 1. Soil Mesture Sensor FC-28**

Arduino Mega 2560 is a microcontroller board that uses Atmega2560 microcontroller, Arduino Mega 2560 has 54 digital input / output pins, of which 15 pins can be used as PWM outputs, 16 as analog inputs and 4 pins as UART (serial hardware port), 16MHz crystal oscillator, USB connection, power jack, ICSP header, reset button. This is all that is needed to support a microcontroller. Simply connect it to the computer via a USB cable or power connected to the power supply or battery to start activating it. Arduino Mega 2560 is compatible with most shields designed for Arduino Duemilanove or Arduino Diecimila. Arduino Mega 2560 is the latest version which replaces Arduino Mega (see Figure 2).

![Arduino Mega 2560](image2)

**Figure 2. Arduino Mega 2560**

In designing the Portable Farming system deliberately used Arduino Mega 2560, this is by considering the most number of pins compared to other Arduino. This Portable Farming requires at least 12 digital pins to connect to Soil Moisture Sensor FC-28 and 18 digital pins for three Motor Drivers that are used to drive the three stepper motors, so that the Arduino Mega 2560 is very qualified for the application of a microcontroller on the Portable Farming system (see Figure 3).
The final step after all the devices are available, now is the time to design the movement of the stepper motor so that it can move at a certain distance both horizontally and vertically (see Figure 4).

Before we design a Portable Farming system, we need to determine the good distance between plants so that the productivity of agricultural land can be produced optimally. Spacing is the amount of space between plants and rows of plants. Spacing is good when planting directly in the field or in containers or polybags. Based on literature, the distance between plants is around 30cm or the area of land for one plant is around 900cm², so that the planted area is around 90cm×120cm or around 10,800cm² [13]. Figure 4 is a tool design, it can be drawn that the position of the stepper motor is placed in a position of 15cm×15cm (center position for each crop land), because it forms a 3×4 matrix, the motor will move at a distance of 15cm, 45cm and 75cm for horizontal movement, while for vertical stepper motors will move in the position of 15cm, 45cm, 75cm and 105cm, thus we can make a mapping of the movement of the stepper motor in a matrix table stepper motor 3×4 (see Table 1).
Next, we make a gear with a circle length of 1 cm assuming it moves for 1 second, so both horizontal and vertical stepper motors will move 1 cm for 1 second. So that tables can be made like Table 2.

### Table 2. Motor Movement Time

| Plant Location | Horizontal | Vertical |
|----------------|------------|----------|
| Plant 1        | 15 seconds | 15 seconds |
| Plant 2        | 45 seconds | 15 seconds |
| Plant 3        | 75 seconds | 15 seconds |
| Plant 4        | 15 seconds | 45 seconds |
| Plant 5        | 45 seconds | 45 seconds |
| Plant 6        | 75 seconds | 45 seconds |
| Plant 7        | 15 seconds | 75 seconds |
| Plant 8        | 45 seconds | 75 seconds |
| Plant 9        | 75 seconds | 75 seconds |
| Plant 10       | 15 seconds | 105 seconds |
| Plant 11       | 45 seconds | 105 seconds |
| Plant 12       | 75 seconds | 105 seconds |

Thus if one day the Soil Moisture Sensor FC - 28 detects soil moisture below the specified value, for example sensor 5 for plant 5 then the stepper motor will move 45 cm horizontally with a travel time of about 45 seconds and 45 cm vertical direction with a travel time of about 45 seconds. In other words the stepper motor will be in plant 5 with a travel time of about 90 seconds from the starting / starting point (standby position). To get the accuracy of the system work in giving water and liquid fertilizer, the system is equipped with a sower that automatically moves according to sensors that detect plants to be given watering or liquid fertilizer. For this reason, a library L298N.h is needed to regulate the movement of the stepper motor in accordance with the position of a sensible humidity sensor that detects that a plant needs watering or liquid fertilizer.

From Tables 1 and 2 and also the library L298N.h where from the commands contained in the library such as driver.forward (speed, time delay); where the speed is a variable with values 0 - 255 and time delay with units of milliseconds, for example driver.forward (150.1000); meaning that the stepper motor will move as far as 150 cm per second, because in the initial design we have determined that the length of the gear circle for the stepper motor is 1 cm. After the movement of the stepper motor is smooth, then we determine the travel time of about 1 minute (30000 milliseconds in the horizontal direction and 30000 milliseconds in the vertical direction).
milliseconds in the vertical direction), so if the stepper motor moves to plant position 5 then we can give the `driver.forward` command (45, 30000); whereas to return to the initial position with the `river.backward` (45,30000) command; so we can make a table like Tables 3 – 6.

### Table 3. Driver Forward Horizontal

| Plant Location | Horizontal          |
|----------------|---------------------|
| Plant 1        | `driver.forward(15,30000);` |
| Plant 2        | `driver.forward(15,30000);` |
| Plant 3        | `driver.forward(15,30000);` |
| Plant 4        | `driver.forward(45,30000);` |
| Plant 5        | `driver.forward(45,30000);` |
| Plant 6        | `driver.forward(45,30000);` |
| Plant 7        | `driver.forward(75,30000);` |
| Plant 8        | `driver.forward(75,30000);` |
| Plant 9        | `driver.forward(75,30000);` |
| Plant 10       | `driver.forward(105,30000);` |
| Plant 11       | `driver.forward(105,30000);` |
| Plant 12       | `driver.forward(105,30000);` |

### Table 4. Driver Forward Vertical

| Plant Location | Vertical          |
|----------------|-------------------|
| Plant 1        | `driver.forward(15,30000);` |
| Plant 2        | `driver.forward(45,30000);` |
| Plant 3        | `driver.forward(75,30000);` |
| Plant 4        | `driver.forward(15,30000);` |
| Plant 5        | `driver.forward(45,30000);` |
| Plant 6        | `driver.forward(75,30000);` |
| Plant 7        | `driver.forward(15,30000);` |
| Plant 8        | `driver.forward(45,30000);` |
| Plant 9        | `driver.forward(75,30000);` |
| Plant 10       | `driver.forward(15,30000);` |
| Plant 11       | `driver.forward(45,30000);` |
| Plant 12       | `driver.forward(75,30000);` |

### Table 5. Driver Backward Horizontal

| Plant Location | Horizontal          |
|----------------|---------------------|
| Plant 1        | `driver.backward(15,30000);` |
| Plant 2        | `driver.backward (15,30000);` |
| Plant 3        | `driver.backward (15,30000);` |
| Plant 4        | `driver.backward (45,30000);` |
| Plant 5        | `driver.backward (45,30000);` |
| Plant 6        | `driver.backward (45,30000);` |
| Plant 7        | `driver.backward (75,30000);` |
| Plant 8        | `driver.backward (45,30000);` |
| Plant 9        | `driver.backward (75,30000);` |
| Plant 10       | `driver.backward (105,30000);` |
| Plant 11       | `driver.backward (105,30000);` |
| Plant 12       | `driver.backward (105,30000);` |
Table 6. Driver Backward Vertical

| Plant Location | Vertical                  |
|----------------|---------------------------|
| Plant 1        | driver.backward (15,30000); |
| Plant 2        | driver.f.backward (45,30000); |
| Plant 3        | driver.f.backward (75,30000); |
| Plant 4        | driver.backward (15,30000); |
| Plant 5        | driver.backward (45,30000); |
| Plant 6        | driver.backward (75,30000); |
| Plant 7        | driver.backward (15,30000); |
| Plant 8        | driver.backward (45,30000); |
| Plant 9        | driver.backward (75,30000); |
| Plant 10       | driver.backward (15,30000); |
| Plant 11       | driver.backward (45,30000); |
| Plant 12       | driver.backward (75,30000); |

The last step after the system runs as desired is to connect Arduino with the WiFi module ESP - 01. In the study using the BLYNK application which is a free application and does not consume a lot of data plan, besides that this application is equipped with some good features. To connect this application simply add the blynk.h library and the BLYNK application installed on the smartphone (see Figure 5).

![Figure 5. BLYNK Sketch Application](image)

By using the BLYNK application that has been installed on a smartphone, we can monitor the condition of our farmland whenever and wherever we are, to activate this application can access via email that was registered earlier when the application is embedded into Arduino (see Figure 6).
3. Results and Discussion

Portable Farming is designed with the following specifications: (1) Length of planting land 120cm, (2) Width of plant land 90cm, (3) Material used to frame steel and wood, (4) Tool design is made like a wheel to make it easy to carry anywhere (portable), (5) Equipped with 3 Stepper Motor and 12 Soil Moisture Sensors (soil moisture sensor), (6) Containers for portable land made of acrylic so that the ground does not make direct contact with a frame made of steel, this is certainly to avoid corrosion due to the watering process and applying liquid fertilizer to the plant.

After the tool assembly is complete, it is time for the tool to be implemented by referring to several calculation tables (Table 1 to Table 6). Test these tables to synchronize whether the libraries that we use in the design of the tool can be proven to run the stepper motor in accordance with Table 3 to Table 6. For that to prove the movement of the stepper motor to fit the calculation concept that has been done we try to do a case study, for example the humidity sensor detects 3 agricultural land such as plant 3, plant 8 and plant 12.
### Table 7. Case study test results for plant 3, plant 8 and plant 12.

| Stepper Motor Location | Horizontal | Vertical |
|------------------------|------------|----------|
| T0 – T3                | 75cm/75s   | 15cm/15s |
| T3 – T0                | 75cm/75s   | 15cm/15s |
| T0 – T8                | 45cm/45s   | 75cm/75s |
| T8 – T0                | 45cm/45s   | 75cm/75s |
| T0 – T12               | 75cm/75s   | 105cm/105s |
| T12 – T0               | 75cm/75s   | 105cm/105s |

The case study shows that libraries to support the movement of stepper motors are indeed proven to be able to do the movements in accordance with the instructions that have been given and of course this is useful for making more complex movements such as large agricultural areas. Moreover, with the support of IoT, farmers can monitor remotely the condition of agricultural land without the need to be preoccupied with conducting site inspections. For the movement of the stepper motor, it is regulated that the position of the motor is always in the stand position (T0) and will always return to the position after the motor has made the move, this is done according to the design of the system that was made with the aim that in addition to avoiding that the stepper motor is not exposed to rain during the rainy season, so that the stepper motor used is much more durable.

From Table 7 we can calculate the travel time and the distance for watering or applying fertilizer to plants 3, plants 8 and plants 12. For plants 3 (T0 - T3 - T0) stepper motor moves 180cm with a travel time of 180 seconds, for plant 8 (T0 - T8 - T0) stepper motor moves 240cm with a travel time of 240 seconds and for plant 12 (T0 - T12 - T0) stepper motor moves 360cm with a travel time of 360 seconds. Thus to conduct a case study of watering plants 1, plant 8 and plant 12 takes about 780 seconds with a distance of 780cm. For the concept of vast agricultural land, this system can be developed by using drones to water and fertilize plants, where the drones can reach a large area of planted land up to the size of hectares (1000m²).

### 4. Conclusion

By using appropriate technology, portable agricultural land can be made with a size that can be adjusted to the area of land / yard of each community, especially urban communities. Even with this technology that starts to spread to IoT, we can apply the concept of IoT by using the help of an Android application in the form of BLYNK where this application does not take up a lot of data plan. This application is appropriate to be applied to agricultural land in urban communities that can turn home yards into land moveable agriculture so that you can save a limited yard area, especially for urban communities. Another advantage without us realizing this concept can change the yard into a Green House that is environmentally friendly.

### Acknowledgement

The author would like to thank the Computer System Study Program at the Universitas Komputer Indonesia, Study Program in Agrotechnology, Agricultural Cultivation, Faculty of Agriculture, University of Padjadjaran in particular to Prof. Dr. Ir. H. Denny Kurniadie, M.Sc who was helpful to the author in making this Portable Farming design.

### References

[1] Dagar, R., Som, S., & Khatri, S. K. 2018. Smart Farming–IoT in Agriculture. In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 1052-1056). IEEE.

[2] Zheng, L., Li, M., Wu, C., Ye, H., Ji, R., Deng, X., ... & Guo, W. 2011. Development of a smart mobile farming service system. *Mathematical and computer modelling*, **54**(3-4), pp. 1194-1203.

[3] Amandeep et al., “Smart farming using IOT,” in 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2017, 2017.

[4] Kyaw, T. Y., & Ng, A. K. 2017. Smart aquaponics system for urban farming. *Energy Procedia*, **143**, pp.
342-347.

[5] Grogan, A. 2012. Smart farming. Engineering & Technology, 7(6), pp. 38-40.

[6] Gondchawar, N., & Kawitkar, R. S. 2016. IoT based smart agriculture. International Journal of advanced research in Computer and Communication Engineering, 5(6), pp. 838-842.

[7] Pomalingo, M. F., Botutihe, S., & Ginting, A. S. 2018. Pengujian Karakteristik Fisik Jagung Bisi 2 untuk Mendesain Mesin Pemipil Jagung Portabel. Jurnal Technopreneur (JTech), 6(2), pp. 36-40.

[8] Husdi, H. 2018. Monitoring Kelembaban Tanah Pertanian Menggunakan Soil Moisture Sensor Fc-28 dan Arduino Uno. ILKOM Jurnal Ilmiah, 10(2), pp. 237-243.

[9] Eller, H., & Denoth, A. 1996. A capacitive soil moisture sensor. Journal of Hydrology, 185(1-4), pp. 137-146.

[10] Robinson, D. A., Campbell, C. S., Hopmans, J. W., Hornbuckle, B. K., Jones, S. B., Knight, R., ... & Wendroth, O. 2008. Soil moisture measurement for ecological and hydrological watershed-scale observatories: A review. Vadose Zone Journal, 7(1), pp.358-389.

[11] Davis Instruments, “Soil Moisture Sensor 6440,” Thames Val., 2007.

[12] Yahwe, C. P., Isnawaty, I., & Aksara, L. F. 2016. Rancang Bangun Prototype System Monitoring Kelembaban Tanah Melalui Sms Berdasarkan Hasil Penyiraman Tanaman “Studi Kasus Tanaman Cabai dan Tomat”. semantIK, 2(1).

[13] Wasonowati, C. 2011. Meningkatkan pertumbuhan tanaman tomat (Lycopersicon esculentum) dengan sistem budidaya hidroponik. Agrovigor: Jurnal Agroekoteknologi, 4(1), pp. 21-27.