2nd International Conference on Advanced Information Technologies (ICAIT), Nov. 1 - 2, 2018, Yangon, Myanmar

Development of a Software Program for Automatic Cartesian Farming Robot

[1]Jedsada Tangmongkhonsuk, [2]Abhishek Meena, [1,3]Pruk Sasithong, [1]Sanika Wijayasakra, [1,3]Gridsada Phanomchoeng, [1,3]Chairat Phongphanphanee, Pisit Vanichchanunt[4] and [1,3]Lunchakorn Wuttisittikulkij

[1]*Faculty of Engineering, Chulalongkorn University*

[2]*Faculty of Engineering, Indian Institute of Technology Ropar*

[3]*Smart Wireless Communication Ecosystem Research Group, Chulalongkorn University*

King Mongkut's University of Technology North Bangkok

Email: gridsada.phanomchoeng@gmail.com, jedsada.tang@gmail.com

Abstract

This paper presents the development of a software platform for our automatic Cartesian farming robot hardware prototype, which is capable of growing multiple types of crops in the soil bay. Seeding and plant watering are carried out automatically and efficiently throughout the crops life cycle virtually without using human labor. Users can design their own plan of growing crops in the farming space through implemented software application from anywhere, anytime using a computer notebook or desktop. Commands from users are then sent among sever which is the Raspberry Pi board and client desktop or laptop device using G-code. This prototype is particularly useful for a home or a small organization to grow crop with more controlled environment and careful management.

Key Words: Automation, Farming Robot, G-code

1. Introduction

Urbanization refers to the population shift from rural to urban residency, the gradual increase in the proportion of people living in urban areas, and the ways in which each society adapts to this change. It is predicted that by 2050 about 64% of the developing world and 86% of the developed world will be urbanized [1].

The rapid economic growth has influenced the larger number of populations to shift from rural to urban residency which is referred to as urbanization. In 2030, the number of urban areas will increase about 60% [1] and this change will create a numerous problem to life style of human being such as urban poverty, urban food insecurity, changes in the climate, crisis in natural resources and many more.

Urban agriculture gives an excellent strategy to reduce urban poverty and food insecurity and further develop the urban environmental management. Urban agriculture plays an important role in enhancing urban food security since the costs of supplying and distributing food to urban areas based on rural production continuous to increase and do not satisfy the demand, especially of the poorer sectors of the population. Therefore, it is especially required use the technology to have a small-scale modular which can be use in an indoor environment to control the growing environment.

FarmBot [2], a commercially available robotic open hardware system, offers a new way to grow multiple types of crops in a small area. Seeds are precisely planted in the soil and crops are automatically watered depending on their needs while the weeds are constantly eliminated. To control and configure FarmBot, there is a free FarmBot web application at my.farm.bot. Thousands of people around the globe have grown crops in their backyards using FarmBot systems.

In our previous work [3], we have built our own automatic Cartesian farming robot hardware prototype that works with the cloud-based software platform of FarmBot. Although the FarmBot can be controlled from web browser using a PC, in some scenarios it is not possible to provide a continue internet connection to the FarmBot. Then it doesn't receive any commands from the FarmBot web application and would not be able to follow the given commands by the user [4]. The main aim of this development project is to build software platform using python from the scratch which is compatible with our FarmBot hardware platform. By using this platform, it will be convenient to control the FarmBot for developers who want to use it in an offline.

The rest of the paper is structured as follows. Section 2 describes overall system and Section 3 explain about implemented software in detail. In the Section 4 we discussed the obtained results and we conclude our work in Section 5.

2. Overall system

In this section the overall hardware prototype of FarmBot system is given in detail. This prototype can be divided into three main components namely, mechanical, electrical and control system. As shown in Figure 1, the FarmBot prototype uses in this work is in 1 meter in length, 0.7 meter in width and about 0.8 meter in overall height. The mechanical system includes three movable axes consist of X, Y and Z axes. X and Y axes use a mechanical system called “Pinion Belt” which can use to fix the belt with the track and allow motors move with the construction along the track. For Z axis the “Lead Screw” mechanism is used which allows tools to move precisely. At the lower end of the Z axis, the universal tool mount is fixed. Our tool set consists of a seeder tool and a watering tool. It can be moved around the farming area according to the control order received.

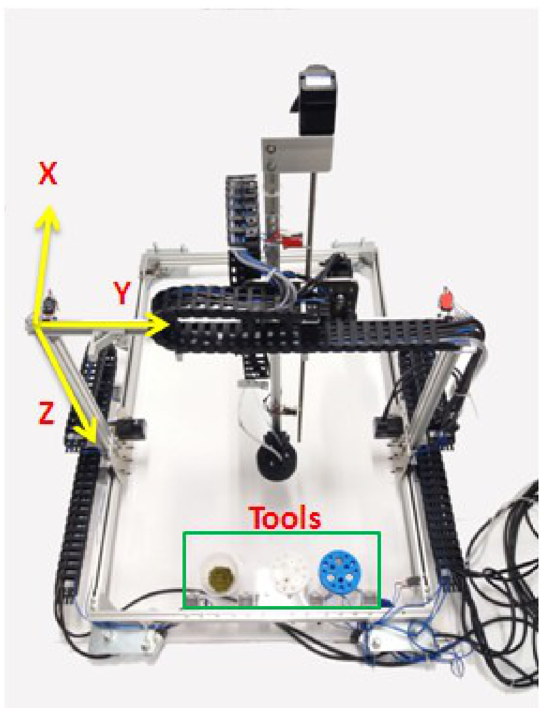

Figure 1. Cartesian farming robot prototype

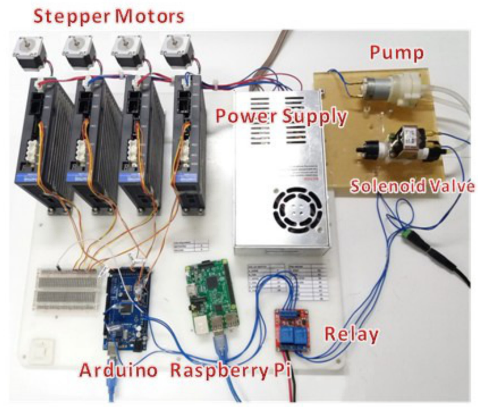

Figure 2. Control system hardware for FarmBot

For the lower level control, an arduino board is installed with the FarmBot using open-source arduino firmware [5]. This is written in C++ language and use to control the motors for movement and the tools for crop-growing functions.

As shown in Figure 2, an Arduino board Mega 2560 is responsible for the simultaneous control of all four linear bipolar motor drivers, thus the universal tool mount can be moved to any desired location with any tools including the watering tool and the seeder. In addition, the Arduino board is used to control two normally close solenoid valves for the watering tool and a vacuum pump for the seeder.

Figure 3, show the arduino receives the commands from the PC application in the form of G-code through a USB cable from a Raspberry Pi controller. The commands were sent through the socket programing [6]. Some commands can be automatically modified by the decision support system before being executed.

3. Software for FarmBot

In our FarmBot program, we make a connection between PC and FarmBot using socket programming in order to create a communication between them. This includes a server and a client as its two main sections where the server listens to the specific port and IP to which clients are connected.

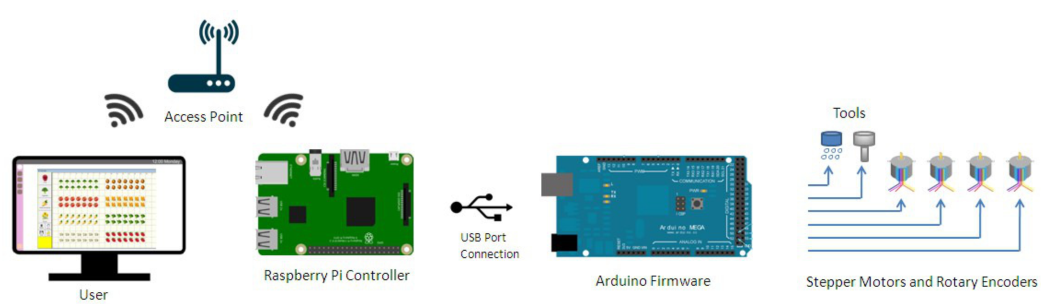

Figure 3. FarmBot communication system

Raspberry-pi (as server):

In our work, the Raspberry-pi is working as a server. Clients can connect to IP of this Raspberry-pi in order to communicate. The raspberry-pi sends commands to the Arduino which is responsible for motions of the FarmBot's motors.

Firstly, the program creates a connection between Raspberry-pi and Arduino where Raspberry-pi and Arduino communicates through serial-communication. Therefore, it adds a python serial library first and next defines Arduino's port and makes a connection.

Secondly, the Raspberry-pi is defined as a server with its IPv4 address. Then, the program has to bind this server to its IP address and port 8000. The server can listen to a number of clients at a time. Thus, the server is ready to accept commands from client. Then, using a loop we make our server to receive commands from clients iteratively otherwise the server can take only one command and get close. Noted it is using the "UTF-8" encoding format for inputs.

Client:

Any client can connect to this server using the IP address and the port number. Once connection is established, client is ready to send the signals to the Raspberry-pi over the local network.

GUI (Graphical User Interface):

To control FarmBot, the program needs a GUI system which is known as graphical user interface. We used python coding language, since a huge number of GUI frameworks/toolkits are available in it. For example, TkInter [5] is GUI programming toolkit. This is most commonly used one to create GUI applications. As Shown in Figure 5, our GUI has a base window which refers to our virtual-farm and this window can be resized in the ratio of actual farm's length and width and grid lines on it gives an idea of actual length (C).

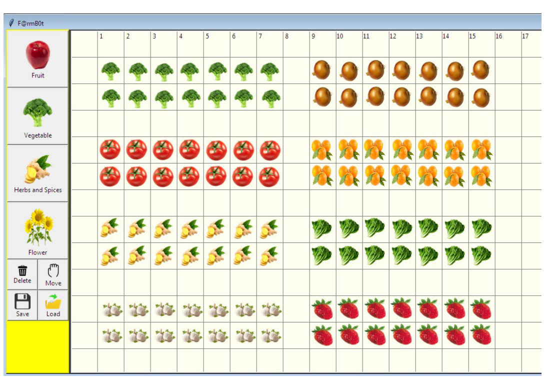

Figure 5. GUI of FarmBot window

As given in Figure 5, the menu bar (A1) on the left side of the window gives different classes of crops, fruits, vegetables, flowers, herbs and spices. User can choose any class based on his desire by just clicking on the button with the images of all famous crops. (A2) Using our application, the user can select the all major fruits, vegetables, flowers, herbs and spices which can be planted in the farm.

Further, our software application provides the options of delete, move, save and load to customize farming plan (B). Therfore, by using the given crops information and the commands the design of the farming plan and the control of the FarmBot can happen automatically. The overall process of Farmbot program is show in the class diagram as Figure 6.

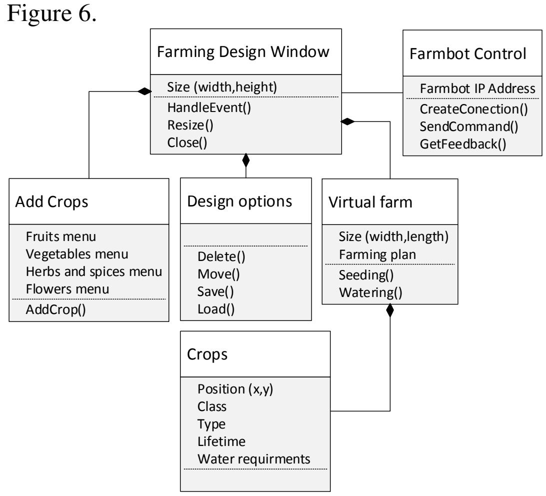

Figure 6. The class diagram of Farmbot program

4. Results and Discussion

The comparison of FarmBot web application and developed FarmBot program is given in Table 1. Our FarmBot program can work in local network in a situation where the internet connection is not available.

Table 1. Comparison of FarmBot's software

	Open source web application	**Developed FarmBot program**
Connection	Need internet connection	Run in local network
Control	Have some latency to control	FarmBot run commands suddenly
Development	Difficult to understand source code for modifying	Convenient to develop additional function in future

Further, it can run commands quickly since the commands are sent directly from the PC. Moreover, the implemented FarmBot software can be developed to convey the application and future scope for automatic agriculture systems such as new crops for seeding, new tools for planting, new sensors for measuring some parameters and etc.

5. Conclusion

In this paper, a software for an automatic farming Cartesian robot as known as FarmBot has been successfully implemented. Our software consists of two main parts, namely communication and GUI control system. For communication part, we use socket programming to build the connection between user PC and the FarmBot. It is convenient for a user to design farming plan with GUI from our implemented FarmBot program and while it can control the FarmBot automatically in an offline situation too.

As future work, we would like to apply this software platforms for other automatic agricultural systems which can be appropriate with user's farming area. Moreover, there are many features to improve and upgrade in the FarmBot system i.e., smart sensor, image processing and farming data visualization.

References

[1] Smart farming: Hope for the world, opportunity for Thailand [Online], [Accessed: August 2018], Available: https://www.scbeic.com/en/detail/product/2812.

[2] FarmBot Official Website [Online], 2014 [Accessed: August 2018], Available: https://farm.bot/.

[3] Y. Monplub, N. Intaratanoo, C Wichitphan, P. Sasithong, G Phanomchoeng and L. Wuttisittikulkij, "A Prototype of Automatic Cartesian Farming Robot," *Proceedings of ITC-CSCC 2018 Conference* (pp. 482-485), Bangkok, Thailand.

[4] FarmBot web application [Online], 2014 [Accessed: August 2018], Available: https://FarmBot.io/.

[5] FarmBot Arduino controller [Online], 2014 [Accessed: August 2018], Available: https://github.com/FarmBot/FarmBot-arduino-firmware/.

[6] M. Xue and C. Zhu, "The Socket Programming and Software Design for Communication Based on Client/Server," *2009 Pacific-Asia Conference on Circuits, Communications and Systems*, Chengdu, 2009, pp. 775-777.