IoT prototyping using Block based programming: An use case of smart agriculture

Meenaxi M Raikar
School of Computer Science and Engineering
K. L. E. Technological University
Hubballi, India
mmraikar@kletech.ac.in

Prajwal Shavi
School of Computer Science and Engineering
K. L. E. Technological University
Hubballi, India
prajwalshavi21@gmail.com

Neha Patil
School of Computer Science and Engineering
K. L. E. Technological University
Hubballi, India
nehatpatil66@gmail.com

Swati Mudavi
School of Computer Science and Engineering
K. L. E. Technological University
Hubballi, India
swatimudavi@gmail.com

Abstract— **The Internet of Things (IoT) has spread its wings in many sectors, such as autonomous driving, healthcare, agriculture, industry, home automation, the environment, smart grids, retail, and logistics. Building the IoT prototype requires knowledge of different disciplines, namely sensors, database programming, protocols, embedded systems programming, and mobile application development. The block-based programming removes all these dependencies and enables the student to build an IoT prototype. The block-based programming tools are available online as open source, which comes to aid in rapid IoT prototype development at ease.**

In this paper, we present the development of an IoT prototype with the use case of smart agriculture using a block-based programming platform available online. The functionalities provided to the agricultural robot (agribot) are ploughing, sowing, saving the data in the cloud database, and monitoring the temperature and humidity of the farming land.

The results are promising; the developer can rapidly deploy the IoT prototype in different domains such as transportation, energy, cities, and industry.

Keywords— MIT app inventor, Arduino Uno, cloud database actuator, DHT-11, Frizing tool.

I. INTRODUCTION

In agriculture, robots are used for weed control, seeding, and harvesting. The use of robotics and artificial intelligence is also helping farmers with soil analysis and environmental monitoring. In the coming years, there is an expectation of an expanding market for robotics in agriculture applications. The technology assists farmers in increasing their productivity, reducing operations costs, lowering maintenance costs, and increasing their overall development. Multipurpose agricultural robots will definitely be helpful to all farmers across the globe because they will reduce the amount of manpower needed to complete tasks while increasing farmers' efficiency. The contributions presented in the paper are:

1. The block-based programming is used for IoT prototyping for the design of the mobile application to control things.
2. The prototype for smart agriculture is demonstrated for functionalities such as ploughing, sowing, and sensor connectivity to the cloud database, as in fig 1.

II. RELATED WORK

In this section, the related work on block-based programming is presented. The development of user-defined blocks for IoT applications is presented for educational usage in the prototype-building process [1]. In [3], the block program platform architecture and design are presented for the development of embedded system applications. Ardublockly and Node-RED tools are explored in [4] for machine learning with IoT applications. In [5], mobile application design for IoT using the Raspberry pi is presented. Sensors available on mobile phones are used in integration with block programming for the development of IoT applications [7]. Visual programming platform is designed for IoT applications with the use case of home automation [8]. A case study of designing an application using block programming is discussed specifically for Bluetooth-enabled connectivity [9]. The experience of exploring block-based programming for IoT application development using mobile devices is described in [10], including the survey results of using MIT app inventor and Snap4Arduino is presented. The development of collaborative IoT projects is presented by considering the Pre-University students in [11]. In [12], an evaluation of block-based programming tools usage in the teaching-learning process of IoT courses is performed. In [13], the solar powered multitask-performing robot is designed using NodeMCU connecting to the FirebaseDB. In [14], a multipurpose agricultural robot is designed using NodeMCU for internet connectivity. In [15], a survey on smart agriculture using modern techniques such as machine learning, IoT, Unmanned Aerial Vehicles (UAV) / drones, soil sampling, yield monitoring, greenhouse farming, hydroponic, phenotyping, and vertical farming is presented. In [16], the devices connected to the internet are controlled by the software-defined controller for efficient network management.

Fig. 1. Internet of Things for smart agriculture use case.

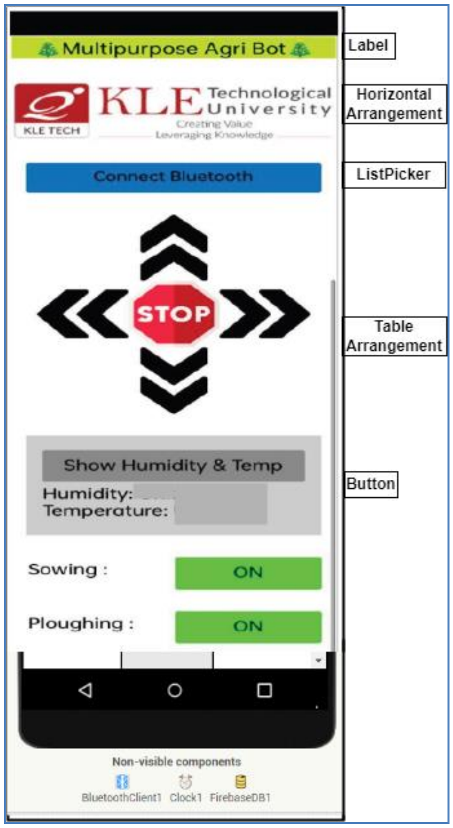

Fig 2. Design of the mobile application.

III. RESEARCH METHODOLOGY

The block-based programming for IoT prototype development enables the developer to deploy the application while quickly learning the different modules. In block-based programming, two main sections are user interface design and block building. The two building blocks are described in the next section.

A. *Designer*

The mobile application developer can design the application's user interface using the designer option. The different options available for designing the user interface are represented as shown in table 1.

1) User Interface: The developer can drag the components on the viewer and design the app as per the user requirements [2]. The user interface for the agriculture purpose is designed as shown in fig 2. The components used for the design of the app are a label, listpicker, and button. The properties of the label are set to display the text and set the background color.

2) Sensor: The various sensor-captured information is processed by the controller for appropriate actuation and decision-making.

3) Connectivity: The developer can drag the non-visible components such as Bluetooth Client, clock, and FirebaseDB, as presented in fig 2. The scan process is initiated when the

Fig. 3. Bluetooth connectivity to control the things.

"Connect Bluetooth" is picked from the list. After the available Bluetooth devices are scanned, a list is generated of nearby devices, as represented in fig 3. One from the list is to be chosen to establish connectivity between the devices. The pairing of the devices is carried out to initiate the data transfer. Once the pairing is successful, data transmission begins between the mobile device and the things.

B. *Blocks*

The different built-in blocks are available, namely control, logic, math, text, lists, dictionaries, color, variables, and procedures [2].

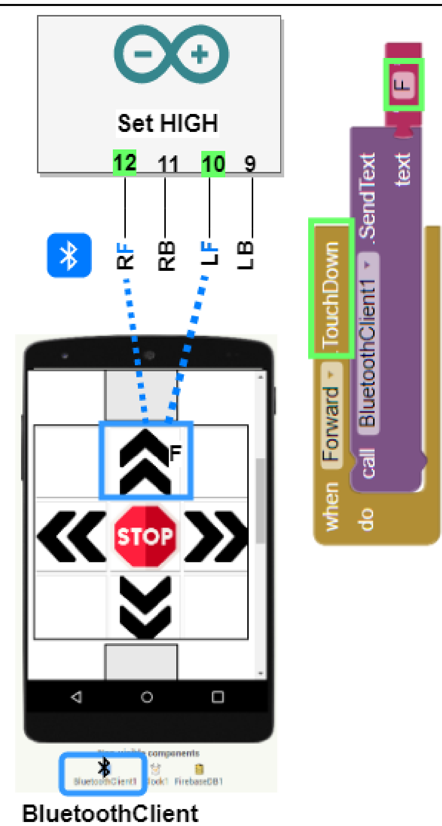

Fig. 4. Bluetooth connectivity enabled block programming for movement of the agribot.

The logic for the block can be provided by specifying the relational and logical operators. The logical operator "=" is used for checking the "ON" or "OFF" status for ploughing and sowing operation, as presented in fig 10. In fig 4, the text block with RED color is used to compare if the forward movement of the bot is to be enabled.

Colors: The different colors can be set to the button using the Colors block. The color green and red is set to the state of ploughing and sowing operation as presented in fig 9. The developer can create a specific color by indicating any value between 0 to 255.

Variables: The initialization of the variables is performed using the set, initialize and get the block. The string is defined to store the temperature and humidity data captured, as presented in fig 10, using the set block.

Procedures: The developer can create the procedures using this block which is PURPLE in color. The procedure is attached to the Bluetoothclient component for sending the text information when the forward, backward, right, left, and stop icons are pressed by the user, as represented in fig 4. The procedure for the display of the sensed data using the DHT-11 sensor is presented in fig 9. The call to ploughing procedure is performed when the button is pressed. The character 'T' is transferred at the press of a button. At the microcontroller program, the ploughing action is performed once the character 'T' is received on pin 7 of the Arduino Uno, as presented in fig 10.

Problem Statement

To design an IoT multi-objective robot for agriculture purposes that can perform ploughing, sowing operations and can also check the temperature and humidity of the surroundings. The hardware requirements for building the agribot prototype are:

- Controller (Arduino Uno Rev3)
- TT Gear motors
- 60 RPM Motor
- Servo motor
- DHT temperature and humidity sensor
- Bluetooth module (HC-05)
- Motor driver (L298N)
- 12V DC Adapter
- 5V DC Adapter

The software requirements for deployment of the model are

- Frizing tool used for drawing the circuit diagram
- Arduino IDE software
- MIT app inventor – online platform for IoT prototype development

IV. IMPLEMENTATION

The CloudIoT architecture is applied to build the IoT prototype of smart agriculture using the block-based programming.

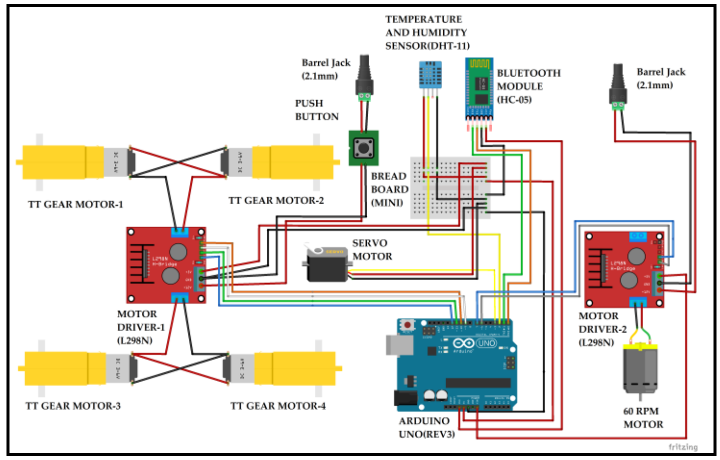

Fig 5. Circuit diagram for the smart agriculture

The circuit diagram for the IoT prototype developed for use case smart agriculture is presented in fig 5. The Arduino Uno is used as the controller to drive the things connected to the network. The Microcontroller is connected and controlled using the mobile app developed with the MIT App Inventory tool. The BlueTooth module HC-05 enables the user having the mobile app to send commands to the agribot.

A. *Agriculture bot movements*

The farmer is able to move the agribot using the mobile application as per table 1.

TABLE 1. MICROCONTROLLER PIN STATE FOR **BOT MOVEMENT**.

Sl No	**PIN**				**Agriculture Bot Movements**
	12	**11**	**10**	**9**	
1	H	L	H	L	Forward
2	L	H	L	H	Backward
3	L	L	H	H	Left
4	H	H	L	L	Right
5	L	L	L	L	Stop

B. *Sowing*

The servo motor is used for generating the "to and for" motion used for sowing the seeds, as presented in fig 8. In [6], the "to and fro" motion is used for opening and closing the switch in the power grid, thereby avoiding electric accidents.

TABLE 2. MICROCONTROLLER **PIN STATE** FOR **SOWING**.

Sl No	**PIN**	**To and Fro Motion**	**SOWING**
	3		
1	60˚	Open	ON
2	0˚	Close	OFF

C. *Ploughing*

Four TT gear motors powered by the motor driver-1 (L298N) are used for the movement of the agribot in different directions as per the user input. The 60-rpm DC motor helps in

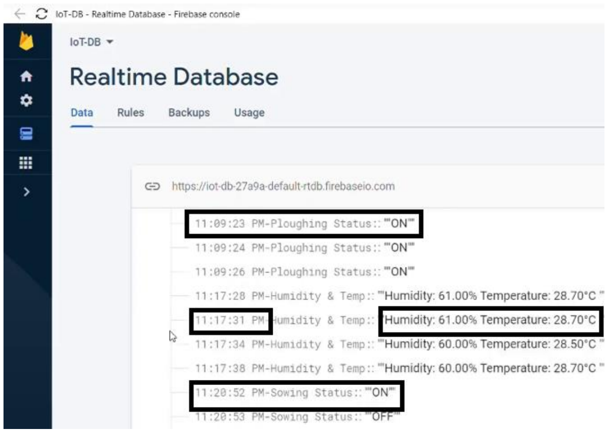

Fig. 6. Real-time storage of IoT data using Firebase.

ploughing operation and is driven by motor driver-2, using the rack and pinion mechanism, as shown in fig 7. In rack and pinion, the circular movement of the gear enables the linear movement of the ploughing rack for it to contact the land. Power is supplied to the whole model using 2 DC adapters, one of 12V and the other of 5V. The 12V DC adapter powers the Microcontroller, Bluetooth module (HC-05), DHT temperature and humidity sensor, and the servo motor used in the sowing operation. The 5V DC adapter power-ups four TT gear motors used for the movement of the bot and also to power the DC motor used in ploughing operation. The procedure for ploughing is presented in fig 10.

TABLE 3 MICROCONTROLLER PIN STATE FOR **PLOUGHING**.

Sl No	**PIN**		**RACK and PINION Movements**	**PLOUGHING**
	7	**6**		
1	H	L	Downwards	ON
2	L	H	Upwards	OFF
3	L	L	Stop	OFF

D. *Environmental sensor information*

The environmental information on agricultural land is captured using the DHT-11 sensor.

Temperature and Humidity: The DHT-11 sensor is used to sense agricultural land's temperature and humidity level.

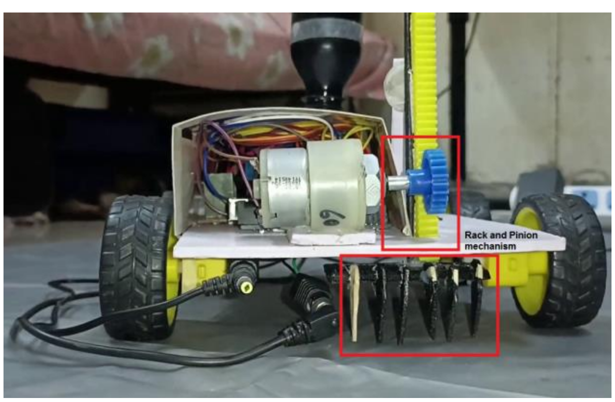

Fig. 7. Ploughing operation using the rack and pinion operation.

E. *Microcontroller programming*

The microcontroller is programmed using algorithm-1 to achieve the required functionalities of the agribot.

F. *Cloud database storage*

The real-time sensor data and the status of ploughing and sowing are stored in the Firebase database for the analysis of the agricultural data, as presented in fig 6. The procedure for connecting to FirebaseDB is represented in fig 9.

Algorithm 1 Microcontroller program to control things.

1: Assumption: Bluetooth Connectivity is established using HC-05.

2: **Input:** User interface provided by the mobile app.

3: **Define the Pin** : Temperature/Humidity Sensor (2), Servomotor (3) and Motor Driver – L298N (6, 7)

Initialization: Serial port communication for Bluetooth (HC-05) using Pin 0 – Receive, 1 – Transmit, Baud rate set as 9600. Pin 12, 11, 10, 9, 6, and 7 are set as OUTPUT, and PIN 3 is set to 60 degrees.

Output:

Loop until the user provides the commands

The PIN values are set for forward, backward, Right, Left, and Stop **movements**, as shown in Table 1.

The PIN values are set for the SOWING operation as per Table 2.

For **the PLOUGHING** operation, the PIN values are set as per Table 3.

Display of the temperature and humidity captured by the DHT-11 sensor.

Store the timestamp, temperature, humidity, and status of ploughing and sowing in the **Cloud database**.

V. CONCLUSION

The block-based programming provides a user-friendly platform for the development of the IoT prototype. The block-based programming is applied for the use case of smart agriculture. The prototype developed includes the different components of IoT applications, such as sensor data capture, database and protocol usage, hardware and software integration, actuation, and cloud storage. The use cases developed using block-based programming are industrial automation, smart notice board, smart lock, and remote vehicle tracking.

Fig 8 **Sowing operation using To and Fro motion.**

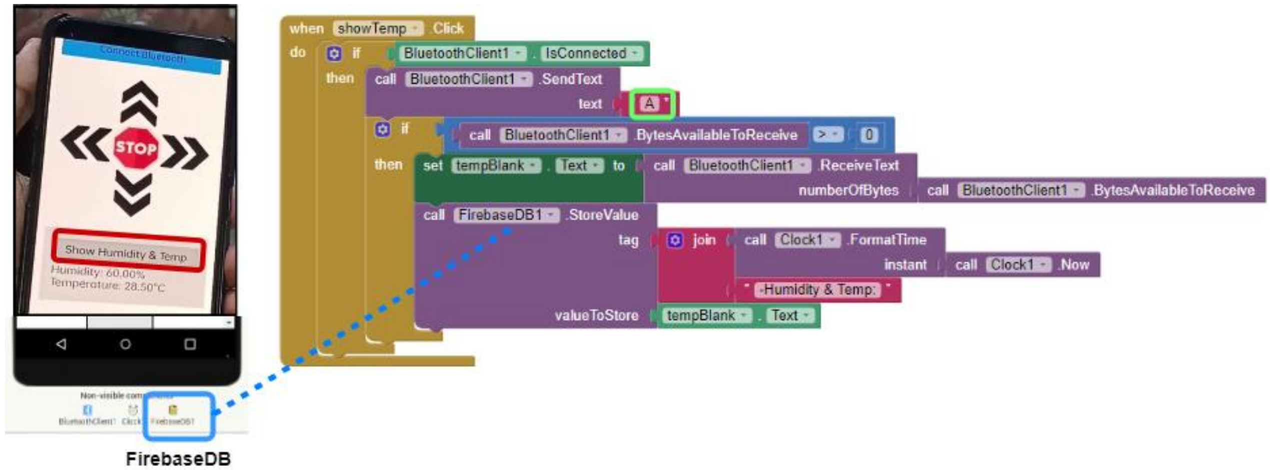

Fig. 9. Block program module for capturing sensor data and store in the Firebase database.

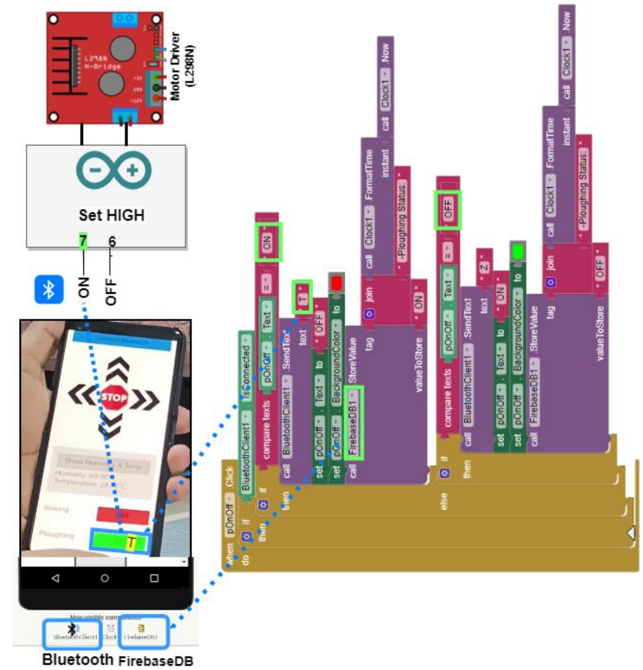

Fig. 10. Block program for plough operation with Bluetooth and Firebase connectivity.

REFERENCES

[1] Y. Tan, M. Rizk, G. Stein and Á. Lédeczi, "User-Extensible Block-Based Interfaces for Internet of Things Devices as New Educational Tools," SoutheastCon 2022, 2022, pp. 711-717, doi: 10.1109/SoutheastCon48659.2022.9763937.

[2] MIT App Inventor, [Online]. Available: http://ai2.appinventor.mit.edu/, Accessed on 06.12.2022.

[3] A. G. D. S. Junior, L. M. G. Gonçalves, G. A. De Paula Caurin, G. T. B. Tamanaka, A. C. Hernandes and R. V. Aroca, "BIPES: Block Based Integrated Platform for Embedded Systems," in IEEE Access, vol. 8, pp. 197955-197968, 2020, doi: 10.1109/ACCESS.2020.3035083.

[4] M. Hauck, R. Machhamer, L. Czenkusch, K. -U. Gollmer and G. Dartmann, "Node and Block-Based Development Tools for Distributed Systems With AI Applications," in IEEE Access, vol. 7, pp. 143109-143119, 2019, doi: 10.1109/ACCESS.2019.2940113.

[5] R. Setiawan, A. R. Anom Besari, I. K. Wibowo, M. R. Rizqullah and D. Agata, "Mobile Visual Programming Apps for Internet of Things Applications based on Raspberry Pi 3 Platform," 2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), 2018, pp. 199-204, doi: 10.1109/KCIC.2018.8628483.

[6] N. Perumalsamy, G. Sivasankar, R. M. Elavarasan, S. Kumar and B. Khan, "Design of Faulty Switching Detection and Alert System to Prevent Fatality of Servicemen During Transformer Maintenance," in IEEE Access, vol. 10, pp. 77362-77374, 2022, doi: 10.1109/ACCESS.2022.3193098.

[7] D. Jean, B. Broll, G. Stein and Á. Lédeczi, "Your Phone as a Sensor: Making IoT Accessible for Novice Programmers," 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1-5, doi: 10.1109/FIE49875.2021.9637272.

[8] N. Bak, B. -M. Chang and K. Choi, "Smart Block: A Visual Programming Environment for SmartThings," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 2018, pp. 32-37, doi: 10.1109/COMPSAC.2018.10199.

[9] T. -L. Chen, I. -A. Su and J. -K. Lee, "Case Study: Design Strategies for Enabling Visual Application Blocks of Bluetooth Library," in IEEE Access, vol. 10, pp. 52630-52654, 2022, doi: 10.1109/ACCESS.2022.3175316.

[10] Ruiz-Rube I, Mota JM, Person T, Corral JMR, Dodero JM. Block-Based Development of Mobile Learning Experiences for the Internet of Things. Sensors. 2019; 19(24):5467. https://doi.org/10.3390/s19245467

[11] S. Trilles, A. Monfort-Muriach, Á. Gómez-Cambronero and C. Granell, "Sucre4Stem: Collaborative Projects Based on IoT Devices for Students in Secondary and Pre-University Education," in IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 17, no. 2, pp. 150-159, May 2022, doi: 10.1109/RITA.2022.3166854.

[12] Sobreira, Péricles de Lima et al. "Usability evaluation of block programming tools in IoT contexts for initial engineering courses." 2020 IEEE World Conference on Engineering Education (EDUNINE) (2020): 1-5.

[13] B. R. Jerosheja and C. Mythili, "Solar Powered Automated Multi-Tasking Agricultural Robot," 2020 International Conference on Innovative Trends in Information Technology (ICITIIT), 2020, pp. 1-5, doi: 10.1109/ICITIIT49094.2020.9071542.

[14] Prajwal Shavi, "Multipurpose Agricultural Bot", https://youtu.be/TbtXa2e0pdk, Accessed on 10.12.2022.

[15] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour and E. -H. M. Aggoune, "Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk," in IEEE Access, vol. 7, pp. 129551-129583, 2019, doi: 10.1109/ACCESS.2019.2932609.

[16] Meenaxi M Raikar, Meena S M, Mohammed Moin Mulla, "Software Defined Internet of Things using lightweight protocol", Procedia Computer Science, Volume 171, 2020, Pages 1409-1418.