



GrowBot: An Educational Robotic System for Growing Food

Lund, Henrik Hautop; Exner, Martin; Jensen, Nikolai Eskild; Leggieri, Massimiliano; Outzen, Malene; Ravn-Haren, Gitte; von Sehested, Malte; Væring, Andreas; Andersen, Rikke

Published in:
Applied Sciences

Link to article, DOI:
[10.3390/app12115539](https://doi.org/10.3390/app12115539)

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Lund, H. H., Exner, M., Jensen, N. E., Leggieri, M., Outzen, M., Ravn-Haren, G., von Sehested, M., Væring, A., & Andersen, R. (2022). GrowBot: An Educational Robotic System for Growing Food. *Applied Sciences*, 12(11), Article 5539. <https://doi.org/10.3390/app12115539>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Article

GrowBot: An Educational Robotic System for Growing Food

Henrik Hautop Lund ^{1,*} , Martin Exner ², Nikolai Eskild Jensen ³, Massimiliano Leggieri ¹, Malene Outzen ⁴, Gitte Ravn-Haren ⁴ , Malte von Sehested ², Andreas Væring ³ and Rikke Andersen ⁴

¹ Center for Playware, Technical University of Denmark, Building 326, 2800 Kongens Lyngby, Denmark; maxleggieri@gmail.com

² Future Classroom Lab, University College Copenhagen (KP), Humletorvet 3, 1799 Copenhagen, Denmark; maex@kp.dk (M.E.); maltex@gmail.com (M.v.S.)

³ Skylab, Technical University of Denmark, 2800 Kongens Lyngby, Denmark; nikomof@gmail.com (N.E.J.); andreas.vaering@gmail.com (A.V.)

⁴ National Food Institute, Technical University of Denmark, Building 201, 2800 Kongens Lyngby, Denmark; maou@food.dtu.dk (M.O.); girh@food.dtu.dk (G.R.-H.); rian@food.dtu.dk (R.A.)

* Correspondence: hhl@playware.dtu.dk

Featured Application: Interactive smart farming educational system for schools.

Abstract: We present the GrowBot as an educational robotic system to facilitate hands-on experimentation with the control of environmental conditions for food plant growth. The GrowBot is a tabletop-sized greenhouse automated with sensors and actuators to become a robotic system for the control of plant's growth. The GrowBot includes sensors for humidity, CO₂, temperature, water level, RGB camera images, and actuators to control the grow conditions, including full spectrum lights, IR lights, and UV lights, nutrients pump, water pump, air pump, air change pump, and fan. Inspired by educational robotics, we developed user-friendly graphical programming of the GrowBots on several means: a touch display, a micro:bit, and a remote webserver interface. This allows school pupils to easily program the GrowBots to different growth conditions for the natural plants in terms of temperature, humidity, day light cycle, wavelength of LED light, nutrient rate, etc. The GrowBot system also allows the user to monitor the environmental conditions, such as CO₂ monitoring for photosynthesis understanding, on both the touch display and the remote web-interface. An experiment with nine GrowBots shows that the different parameters can be controlled, that this can control the growth of the food plants, and that control to make an environmental condition with blue light results in higher and larger plants than red light. Further, the pilot experimentation in school settings indicates that the comprehensive system design method results in a deployable system, which can become well adopted in the educational domain.

Keywords: robotics; education; smart farming; smart greenhouse; food production; comprehensive system design



Citation: Lund, H.H.; Exner, M.; Jensen, N.E.; Leggieri, M.; Outzen, M.; Ravn-Haren, G.; von Sehested, M.; Væring, A.; Andersen, R. GrowBot: An Educational Robotic System for Growing Food. *Appl. Sci.* **2022**, *12*, 5539. <https://doi.org/10.3390/app12115539>

Academic Editor: Antonio Fernández-Caballero

Received: 11 April 2022

Accepted: 24 May 2022

Published: 30 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Most development of robotic systems for education (educational robotics) has aimed at creating systems for teaching programming, robotics, and related technical subjects. Prominent examples of such educational robotics include LEGO Mindstorms, Nao, and Fable robots. These are simple, user-friendly robotic systems which allow students to program the systems to task-fulfilling behaviors with user-friendly graphical programming interfaces, giving the students hands-on insight into the technical subjects related to sensors, actuators, robotics, programming, etc.

Despite the user-friendliness of these educational robotic systems, it can often be challenging for teachers to apply the systems in the school curriculum beyond the technical domain, for instance when teaching about food, healthy eating, and biology. It is the

aim of the work presented here to develop an educational robotic system, which permit students hands-on experimentation with food plant growth through the programming of environmental conditions. The environmental conditions are made fully controllable in a tabletop-sized robotic greenhouse called a GrowBot. The GrowBot is automated with sensors and actuators to become a robotic system for the control of the growth of plants growing in a water solution (hydroponics (With hydroponics, soil borne diseases and pests are avoided, less water for growing is needed, and, in some cases, hydroponics may result in faster growth rates and increased yields [1])). The GrowBot robotic system allows students to parameterize growth conditions for the natural plants in terms of temperature, humidity, day light cycle, wavelength of LED lights, nutrients, etc. Thereby, students can perform experiments to develop a parameterization of the optimal growth setting for different natural plants. Programs with different parameter settings lead to different growth of the food plants in the GrowBot controlled with these programs.

With the GrowBot, we have developed a tabletop-sized smart farm which is directed for educational purposes. The smart farm is defined as “a continuous production system that cultivates crops without the human intervention through the automatic control in a suitable space for the crops growth environments such as light, temperature, humidity, carbon dioxide levels, and nutrients” [2]. Putting emphasis on the comprehensive systems design, we aim at delivering a smart farm system for the educational sector with simplicity and user-friendliness for teachers and pupils, easy access for programming and monitoring, and readiness for implementation in school practices, with educational resources matching the curriculum and learning objectives.

A first, fundamental condition for reaching this goal is to provide a robust hardware and software system, in which the environmental conditions are controllable (and understandable) to provide plant growth. Therefore, the present work puts emphasis on this technical development and verification as the foundation for future implementation in school interventions.

By the focus on comprehensive systems design of the GrowBot with easy operation, user-friendliness, and implementation in the educational sector, the GrowBot takes inspiration from educational robotics, and stands out from related work on smart farming.

2. Related Work

Our first iteration for the development of the GrowBots started with a prototype of the MIT Media Lab food computer [3,4] before we went on to create the GrowBots. The MIT Media Lab Open Agriculture Initiative (OpenAg) projects launched the vision and interest among researchers and communities to create smart urban farming based on food computers that could automate food growth in the local vicinity of the consumer in any (e.g., urban or hostile) environmental condition. OpenAg developed the Personal Food Computer with a desktop size based on the Raspberry Pi 3 and Arduino Mega 2560 to interface to sensors and actuators. The aim was to create a system where users could generate, share, and reproduce so-called recipes of growth conditions, and that could be deployed and used in schools from elementary to high school levels. Indeed, by creating an open architecture, a large international community was created around the OpenAg Personal Food Computer being used and deployed in academic research development and in schools. For example, it served as our starting point for a prototype before designing and creating our GrowBot.

A main idea behind smart farming and the OpenAg food computer is that it should be possible to easily grow food plants and replicate experiments, e.g., when sharing so-called recipes. However, the OpenAg met criticism that it was difficult to make the food computer work (e.g., in Syria), to make plants grow, to achieve robustness of the food computers, and to make them work in school settings [5]. Partly due to such criticism of those Personal Food Computers, the MIT Media Lab had to shut down the project in 2020. With the development of the GrowBots, we aim at addressing the issues that were shortcomings

for the Personal Food Computers for bringing the smart farming approach into use in the educational sector.

There are several other smart hydroponic farming research projects. For instance, Van et al. [6] developed the PlantTalk with automatic lighting, water spray, etc. to communicate and be controlled by a smartphone. Since the aim is to reduce carbon dioxide (CO₂) concentration in indoor rooms, the PlantTalk is not sealed and therefore not controllable in many parameters. Further, PlantTalk seems to have a user interface not designed for novice users. Mehra et al. [7] also built a very simple open system with no sealed chamber using a simple breadboard approach for connecting sensors and actuators to Arduino and Raspberry Pi 3. Interestingly, they tried to use deep neural networks as a machine learning approach to search for appropriate mapping between sensor values and actuator output for the control of the system, though they did not present much data on the appropriateness of the approach. Marques et al. [8] developed the iHydroIoT system as an IoT monitoring system for hydroponics. It allows remote monitoring of sensory data on an iOS device (e.g., iPhone), but it is only a monitoring device and it does not offer control of actuators. Angeloni and Pontetti [9] describe the RobotFarm as an automatic hydroponic greenhouse-appliance to bring cultivation directly to everyone's home. The RobotFarm is imagined to be a built-in appliance in a standard kitchen. Unfortunately, Angeloni and Pontetti [9] provide no technical details of the hardware, no program details, and likewise only little details on the user interface.

Similar to our approach, Stevens and Shaikh [10] created their MicroCEA as an adapted version of the MIT Personal Food Computer, and Boanos et al. [11] also made several changes to the MIT Personal Food Computer to create a cost-effective automated food computer to overcome some of the shortcomings of the MIT Personal Food Computer. Even if little data is provided, Boanos et al. [11] showed indications of differences in growth conditions between blue and red light conditions on cilantro grown inside such a food computer. There was no work on the educational suitability and adoption in these projects.

For the programming of the GrowBot, we found inspiration in related work in educational robotics and graphical programming. For educational technology, programming has evolved into user-friendly graphical programming. In the 1990s, LEGO Mindstorms invented and launched a simple graphical programming interface for children, and other even simpler graphical programming interfaces were developed for children's programming of the LEGO Mindstorms robots [12]. This kind of graphical programming environment inspired most other popular graphical programming interfaces over the subsequent decades. For instance, similar graphical programming interfaces were developed in the form of Scratch [13] and Google Blockly [14] (e.g., used in the Fable robots for school pupils [15,16]). The micro:bit [17] uses a simple graphical block programming interface inspired by this tradition, namely Microsoft Makecode.

3. Design Methodology

For developing and designing the GrowBot, we used our general research and design methodology for creating user interactive hardware (HW) and software (SW) artefacts to be applied in real world settings. The methodology aims at allowing us to provide positive change to the society through engineering. The methodology arises from decades of work in the area of Playware, which is defined as intelligent hardware and software that creates play and playful experiences for users of all ages [18]. Similar to the playware focus on creating intelligent artefacts for playful adoption in real world settings, with the GrowBot we aim to create an impact in the educational domain with a new kind of intelligent hardware and software. Therefore, the methodology is viewed as applicable for the development of the GrowBot.

The methodology is illustrated in Figure 1. It is a comprehensive system design with development areas for hardware development, control software development, user interaction development, application development, and development of domain adoption. This contrasts many other engineering research methods, which only focus on the lower

layers and miss to make a comprehensive design focusing on all aspects needed for creating successful deployment in society. In the comprehensive system design, it is important to identify, develop, and verify the domain issues that leads to the adoption of the system in the particular domain in the society. For instance, in the present case, it is also crucial to focus on the food growth, the educational application, and general school practice for obtaining the wide deployment in schools.

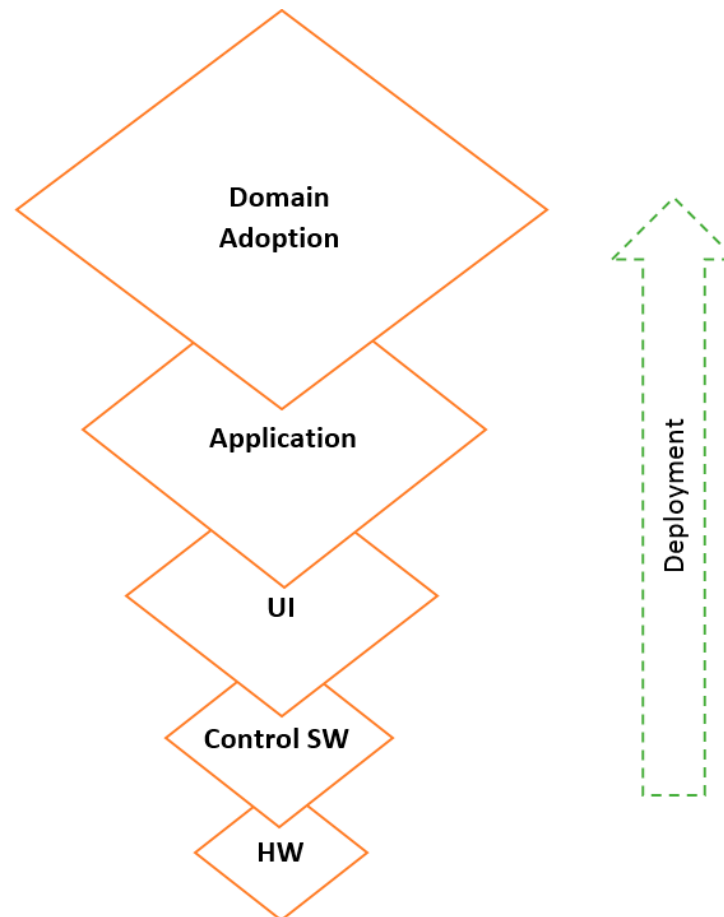


Figure 1. The comprehensive system design method with layers needed to ensure deployment and adoption in society.

Despite the potential perception of the different development areas being discrete processes, it is important to understand that during the development, these areas of development are interlinked and intertwined. In Figure 1, there is a two-way information flow between upper layers and lower layers, so in the development process, for instance the application development process may inform and put requirements on the lower levels of user interaction development and SW and HW development. Indeed, the development process is iterative with iterative technology specification and development, and in upper layers together with people in their real world environment.

For investigating and verifying the developments in this R&D design process, we used a mixed research method, where approaches from the positivistic and interpretivistic epistemologies are merged [19]. There are issues regarding technological stability and robustness for the lower layers in Figure 1, which lends themselves best to experimental research of a positivist nature, whereas issues regarding human interaction for the higher layers in Figure 1 may lend themselves best to participatory research of an interpretivist nature. Therefore, the research method demands that participants build shared knowledge and language about the environment, and research focuses on how results from the two

directions are to be combined. The research method facilitates this combination by focusing on synthesis in the iterative process.

It is of utmost importance for reaching the result of impact in society with deployment of the project development that the process results in robust layers built on top of each other (Figure 1). Higher layers will fail and have little importance if lower layers are not robust. This is well illustrated with the MIT OpenAg Personal Food Computer initiative, which unfortunately failed due to problems with the lower layers of HW and SW robustness and repeatability. Likewise, decades of research in development of intelligent HW and SW artefacts applied in society (e.g., LEGO Mindstorms in play and education, Universal Robots in industrial production, Fable Robots in education and home use, MusicTiles in music production and live performances, Moto Tiles in hospitals and senior care) have shown us that even the slightest glitches in the lower layers of hardware and software control will influence significantly and define the user experiences in higher layers. In the following, we will describe how the layers are developed, aiming at robust layers built on top of each other to ensure ease of future development of the higher layers on top of robust lower layers, and aiming at creating a comprehensive system for a successful real world deployment.

4. Hardware and Software Architecture

We developed and tested the GrowBot hardware in the DTU Skylab and DTU Playware Lab making and assembling the frames and components. The GrowBot measures 450 mm × 450 mm × 600 mm. The GrowBot consists of the chamber and a 150mm top part which holds the electronics, see Figures 2 and 3. At the bottom of the chamber, there is room for a water tray with a lid with 4 × 4 holes in which the plants can grow. The front side of the chamber is a door which can be tightened when closed to ensure an airtight space inside the chamber. On the front of the top part, there is a small 5" touchscreen display and a connector for a micro:bit.



Figure 2. The GrowBot with an open door to the chamber, which contains the water tray at the bottom, and small basil plants in the cups in the tray. Water and fertilizer bottles are mounted on the side.

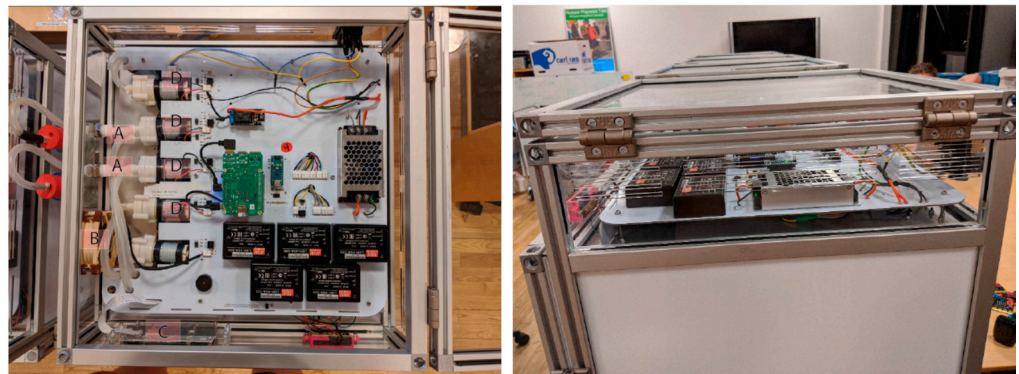


Figure 3. Left: Top view of the GrowBot, and Right: side view of the GrowBot upper layer with the electronics.

There are three electronic boards in the GrowBot:

- The main-board, which is the primary part of the electronics controlling the communication between the elements and the control of the actuators.
- The LED-board, which is the upper wall of the chamber with the LED grow light mounted on it.
- The micro:bit board, which is a simple connection component between the micro:bit and the main board.

On these the main components are mounted, indicated in Table 1, with the GrowBot being controlled by the Raspberry Pi 4 and the Arduino acting as the sensor and actuator interface controller. A number of sensors are mounted to allow monitoring the conditions in the grow chamber and the water tray of the GrowBot. As outlined in Table 2, these include humidity sensor to sense the humidity in the chamber, CO₂ sensor to sense the CO₂ level in the chamber, temperature sensor to sense the air temperature in the chamber, water level sensor to sense the level of water in the tray, and RGB camera to look down from the chamber ceiling onto the growing plants. Table 3 outlines a number of actuators to provide the means to control the grow conditions in the GrowBot. These include full spectrum lights, IR lights, and UV lights, which are mounted on the chamber ceiling, fertilizer pump, which can pump nutrients from a bottle (The users will have added a nutrient composition of their choice to the bottle) into the water tray, water pump, which can pump water from a bottle into the water tray, air pump to pump air to the oxygen disk placed in the water tray, air change pump to change the air in the chamber, air circulation fan to circulate the air inside the chamber, and heater to heat the chamber.

The Arduino, which is the interface to the sensors and actuators, communicates with the Raspberry Pi main processor on a 2-way serial port. Further, the Arduino is attached to a slot connector, which can hold a micro:bit. The main processor (Raspberry Pi) is also connected to a 5" LCD touchscreen, mounted on the front top part of the GrowBot, allowing users to provide input from the touchscreen and to view things such as current sensor values on the touchscreen. Further, the Raspberry Pi is connected to a 4G/LTE module and antenna to provide communication with a remote server when a SIM card is installed in the GrowBot. This provides the means for data to be transferred to and from the remote server.

Another third party hardware used as an option as control device for the GrowBot is the BBC micro:bit. The BBC micro:bit is an affordable pocket-sized computer designed for children's learning. There are more than 4.5 million micro:bit devices in 60 countries among 20 million children [17]. It originates from the BBC UK Make It Digital project of 2016 and aims at fostering digital creativity in the classroom. The micro:bit is based on a Nordic nRF51822 processor with Bluetooth radio communication, an onboard accelerometer and compass, a small LED mesh, two buttons, and I/O channel (see Figure 4). Due to the widespread use and familiarity of the micro:bit among school children, it was chosen as one of the programming interface options for the GrowBot. Indeed, in Denmark, all 4th grade

school classes (children aged around 10 years of age) were given micro:bits over the last year, and have become familiar with its graphical programming interface and functionality.

Table 1. Main components.

Description	Specification
Processor	Raspberry Pi 4 Model B 2 GB
I/O interface processor	16 Mhz Arduino NANO V3.0 Atmega328 UNO IO Shield
Wireless communication	Raspberry Pi 4G/LTE with base HAT and antenna on the Quectel EC25 Mini PCIe 4G/LTE Module
Display	5" Raspberry Pi LCD touchscreen with 800 × 480 resolution and 108 × 64.8 mm display area (SKU DFR0550)
Power	12 V Power supply and a 3.3 V linear regulator (ADP3338)
PWM controller	PCA9685PW PWM Controller
Air circulation fans	405FH DC Axial fans 9 m ³ /h
LED drivers	LDH45 LED driver
Full spectrum light	LM301B white LEDs
IR light	XLamp XP-E2 IR 730 nm
UV light	XLamp XP-E UV 458 nm
Heater	Ceramic (PTC) air heating element
PWM Controller	PCA9685PW, PWM Controller
Pumps	DC 12 V Air Diaphragm Pump 1.5–2 L/Min
Aeration	Oxygen disk
Humidity and temperature sensor	DHT 22 relative humidity and temperature sensor
CO ₂ sensor	MHZ16 infrared CO ₂ sensor
Water level sensor	DFROBOT SEN0193 capacity soil moisture sensor
Camera	Raspberry Pi 4 camera 150 degree wide angle 5 megapixels (OV5647 sensor)
micro:bit	BBC micro:bit board

Table 2. GrowBot sensors.

Sensor	Value	Meaning
Humidity	0.0–100.0	Relative humidity (in percentage) inside the GrowBot chamber
CO ₂	0–5000	CO ₂ level in ppm inside the GrowBot chamber
Temperature	−40.0–80.0	The temperature in C inside the GrowBot chamber
Water level	0.0–100.0	Percentage of how much the water tray is filled
IsMicroBit	0–1	Is the micro:bit connected (1) or not (0)
MicroBit Button A	0–1	Is button A on the micro:bit pressed (1) or not (0)
MicroBit Button B	0–1	Is button B on the micro:bit pressed (1) or not (0)
Clock	00:00–23:59	Timestamp in the format hh:mm
Camera		Colour photo from above every 8 h (Set by default to 8 h intervals. It can be programmed to other time intervals)

Table 3. GrowBot actuators.

Actuator	Value	Meaning
Full spectrum grow light	0–100	Controls the intensity (percentage) of the full spectrum grow light (max. 100 W)
IR grow light	0–100	Controls the intensity (percentage) of the infrared light (max. 40 W)
UV grow light	0–100	Controls the intensity (percentage) of the ultraviolet light (max. 40 W)
Fertilizer pump	0–1000	Amount of nutrients from the bottle to the tray in ml/h
Water pump	0–1000	Amount of water from the bottle to the tray in ml/h
Air pump	0–10,000	Air to the aeration oxygen disk in the water in ml/h
Air change	0–10,000	Air change of the grow chamber in ml/h
Fan (air circulation)	0–1	Setting the two fans Off (0) or On (1)
Heater	0–40	Heating up the chamber to a desired temperature 0–40 °C.

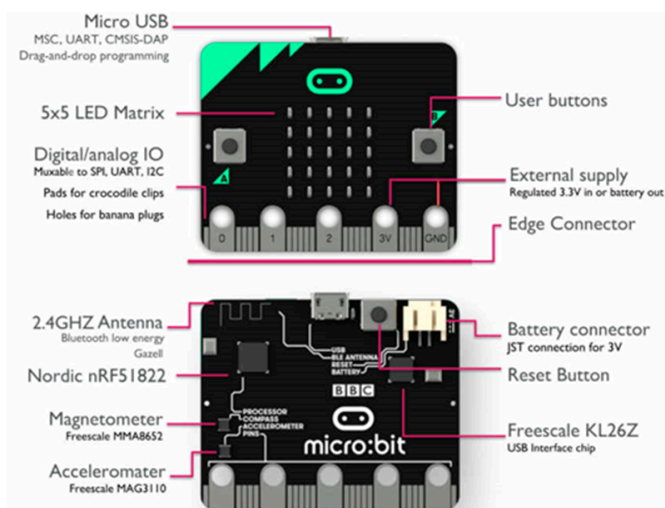


Figure 4. Two of the three programming modalities of the GrowBot. **(Left):** The BBC micro:bit is a small IO module based on the Nordic nRF microcontroller. **(Right):** Programming on the 5" touch display on the GrowBot.

Figure 5 presents the overview of the GrowBot architecture, and how it allows for a seamless interchangeability of different programming and interaction modalities. In the GrowBot, sensors and actuators are connected to the Arduino, which also holds a slot connector (for a micro:bit). There is a 2-way serial port to the Raspberry Pi, to which a touch display and a 4G LTE module is connected. The dotted lines indicate external interaction with the GrowBot provided (1) by inserting a micro:bit, (2) by webserver communication, or (3) by touch on the local display, see Figure 5.

If there is no external interaction, the GrowBot will run with the Raspberry Pi containing a default low level control software, which controls the Arduino access to sensors and actuators of the GrowBot.

The architecture ensures that data can always flow on the solid lines and optionally on the dotted lines. As an example, sensor data can always flow to the display, whereas it optionally flows to the webserver and micro:bit, dependent on whether these are connected or not. When these are connected, the same sensor data can be ported to all connected devices.

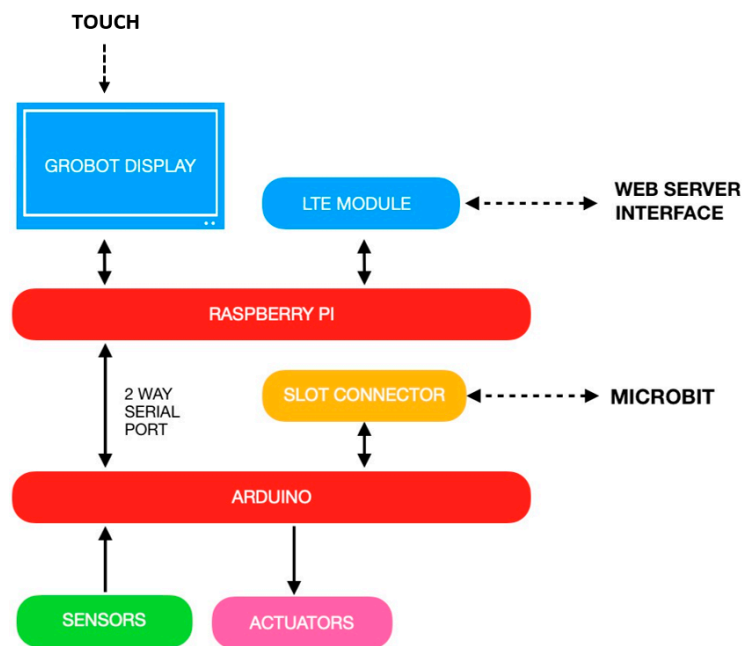


Figure 5. Overview of the GrowBot architecture.

This allows for a control architecture inspired by the subsumption architecture known from behavior-based robotics [20,21]. Here, a number of parallel processes all have access to the sensors and actuators (in contrast with the old-fashioned AI robotics control with one centralized controller). In our case, we may have four control processes accessing the sensor data and controlling the actuators, namely the default (Raspberry Pi) (Potentially, the default behavior could be viewed as part of the Touch display control, but here for the (educational) clarity of the subsumption architecture, we define the default behavior as a separate control of the Raspberry Pi), the Touch display, the webserver, and the micro:bit, as shown in Figure 6. Although it is evident that all control processes can have access to the sensor data in parallel, the other side of the access to the actuators needs more careful coordination, since two processes cannot both control a particular actuator at the same time. Therefore, the behavior-based robotics research has proposed several behavior coordination methods. These behavior coordination methods include suppression, action selection, voting, behavioral fusion (e.g., vector summation), etc. Here, we will find inspiration in the suppression method, which simply is a priority-based coordination that a priori assigns a priority of importance to the different processes. In our case, the processes will have increasing priority from the lowest of default Raspberry Pi up to the highest of the micro:bit in Figure 6.

This means that if there are no other processes running, the default behavior of the Raspberry Pi will be running. If there is input from the Touch display, then this will override the lower level of the default and take control. If there is a webserver input (rule or program with collection of rules), then that will override the lower levels and take control. Finally, if a micro:bit is inserted, it has the highest level of priority, and will override all the lower levels and have control over the actuators.

The advantage of making the control architecture inspired by the subsumption architecture [20] is that one can thereby easily add and remove components, one can develop and refine different components in parallel, and one can even develop and refine some components at a later stage. For instance, the structure of Figure 6 can easily be expanded to include further processes (rules input modalities) with different priority schemes. This provides a generality of the GrowBot, and opens up for multiple user interaction modalities, multiple use cases, and future developments of the GrowBot.

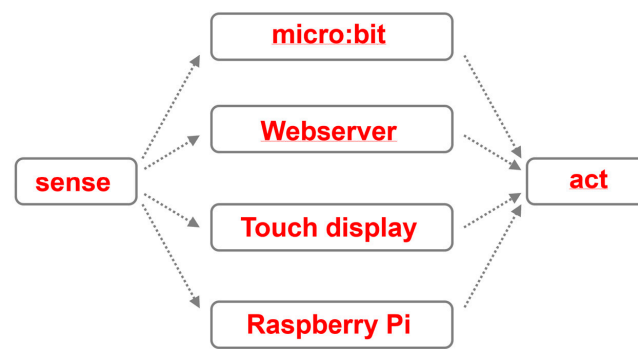


Figure 6. The behavior-based robotics approach is used as inspiration for control of the GrowBot. The suppression method allows a higher level to suppress any lower level in a priority-based coordination.

In the case of the test in schools reported in this paper, the current structure signifies that as soon as the micro:bit is inserted, it has the highest priority of control. Therefore, if the micro:bit is always used, the user will not experience the other options of the GrowBot system.

At the same time, the architecture makes the development more than a tool for the school classes to become a more *general* “GrowBot”. It makes the GrowBot ready to be expanded in future e.g., with multi-user control in a GrowBot Universe via the webserver control option and also via other possible behavior coordination methods such as voting-based coordination, it makes the GrowBot ready to have a smart device control similar to the display control, etc.

In the current implementation, there is a database in the GrowBot (in the Raspberry Pi) which holds the programs and rules (see Figure 7 on the right). A program is a collection of conditional rules and time rules. The programs and rules can be added, removed, and changed by the different user interfaces. In the current implementation of the GrowBot, these interfaces include the Touch Display and Webserver, but the architecture allows the system to easily be extended to include other interfaces for the control of programs and rules in the future.

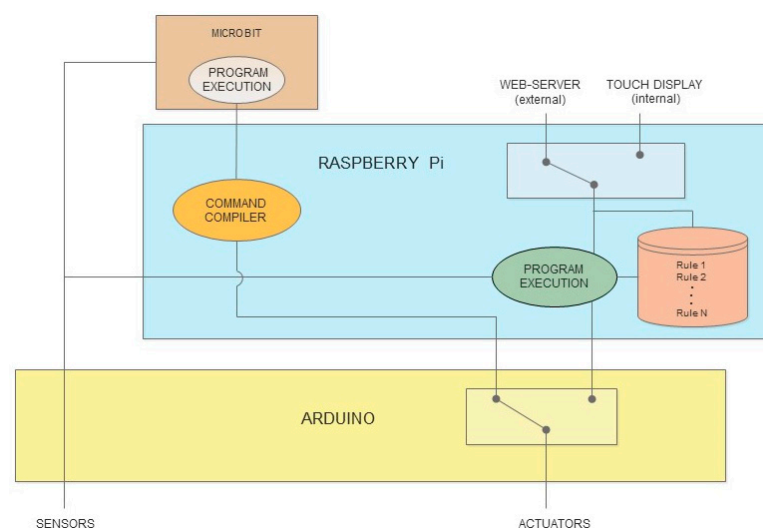


Figure 7. Illustration of the control of the GrowBot, and sensor and actuator communication flow through the Arduino to the Raspberry Pi.

The program execution of the Raspberry Pi will apply the rules written in database for the current program. In the current implementation, a rule is recognized in Algorithm 1 as follows:

Algorithm 1. Example of a rule. This rule sets the water pump to 30 if the moisture sensor value is below 20

```

→ {
  → "sensor": "moisture",
  "condition": "less-than",
  "sensVal": 20,
  "actuator": "water-pump",
  "actVal": 30
→ }

```

When a particular condition is met, an actuator action is applied. In the rule above, this means that when the moisture sensor value is read to be below 20, then the water pump is actuated with the value of 30. The program file contains a set of rules like this that satisfy a particular plant growth criteria.

Initially, a default set of rules is implemented as the *default program* in the Raspberry Pi.

For a fast development in the project, and to allow easy control with external devices, the GrowBot system also allows external devices to directly access sensors and send commands to the actuators. Sending commands to actuators is done through a command compiler on the Raspberry Pi, resulting in a shared actuator command structure between control arising from program execution on such external devices and the internal program execution of the Raspberry Pi (see Figure 7).

When connected to the GrowBot, the micro:bit actuates a sensor switch so the system is informed that the micro:bit has control. A micro:bit command compiler (github plugin) was developed to integrate some custom components that can be used into the MakeCode. These components write to the serial port according to the priority-based coordination, directly bypassing the internal program execution of the GrowBot.

Both the internal program execution and the external program execution (in the micro:bit) control the actuators, which is mainly done on the Arduino serial port. Sharing the same way to write to the serial makes the devices fully compatible and interchangeable.

5. User Interface

As described above, the user can interact with the GrowBot, e.g., by designing rules and programs through the touch display, the webserver, and the micro:bit. In the current experiment for school pupils, the main user interface is provided by the programming of the micro:bit and reading sensory data from the growth process on the touch display and on the webserver. The programming is performed in a graphical programming interface inspired by earlier education technology graphical programming, namely Microsoft Makecode. This makes it easy for school children who might have experience with, e.g., LEGO Mindstorms or Scratch to program the GrowBot, and for others to quickly learn the programming.

On a laptop, in the graphical block programming environment Microsoft Makecode, the school children develop their code for the micro:bit, compile the code and flash the code to the micro:bit, which at this point is USB connected to the laptop. Next, they can detach the micro:bit from the laptop and insert it into the GrowBot in the connector installed on the front top of the GrowBot. The GrowBot will read the code, signal on the display that the micro:bit is in control, and the GrowBot will be controlled by the micro:bit code.

For allowing the GrowBot to be controlled by the micro:bit, we defined and developed a number of blocks for the Makecode environment. These blocks, called 'Groblocks', can be imported in the Makecode environment. They allow the user to control with the various sensors and actuators of the GrowBot. An example of a simple control program for the micro:bit made in the Makecode environment can be seen in Figure 8. In this case, at the start, the air change of the chamber and the air pump for the oxygen disk is set to 1000 mL/h and the fans are set to on (1). A time condition is made so that during night between 22:00 and 4:00, the IR light and the full spectrum light are set to 0, whereas in the other time interval during the day, the IR light is set to 40, and the full spectrum light is set to 16.

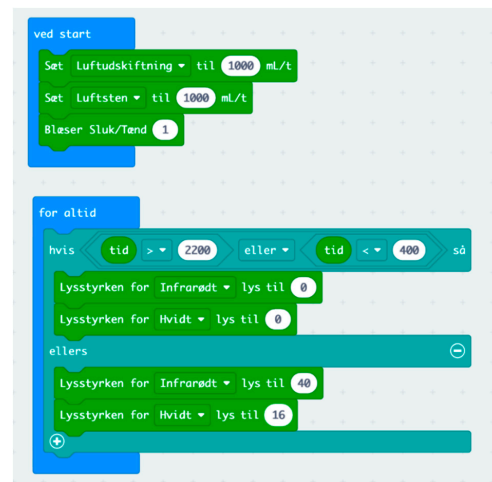


Figure 8. Example of a control program for the GrowBot made in Makecode (in Danish). This code will be compiled and transferred to the micro:bit, which is then placed in the connector on front of the GrowBot.

To allow easy and fast control of the GrowBot, we also developed the option to interact with the GrowBot through the touchscreen display mounted on the front of the GrowBot. Here it is possible to enter conditions and time rules similar to those used in the Makecode environment and allow the GrowBot to be directly controlled by these rules entered on the touchscreen. This is a fast option to make rules, though the small 5" touchscreen may pose some challenges to large fingers.

Finally, the GrowBot provides the option to develop rules and programs on the webserver so that the user can program and edit the control programs remotely on the webserver and then, from the remote access, transfer these for control on the physical GrowBot. The remote access control will be enhanced further in future developments of the GrowBot. Figure 9 shows the simple webserver rules editor with a number of time rules.

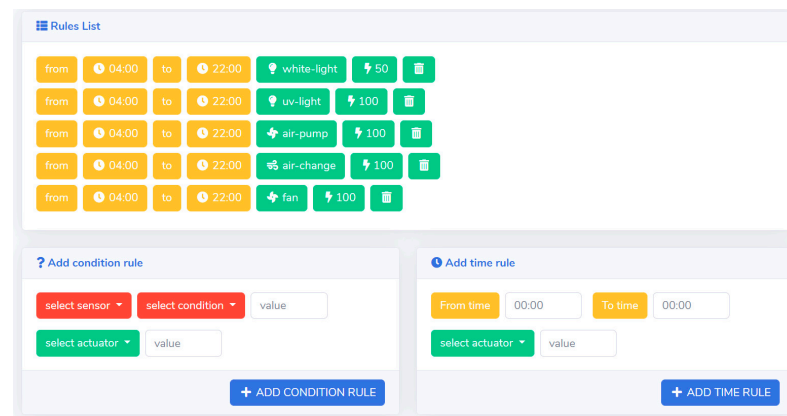


Figure 9. The rules editor on the webserver to make rules, which together form a recipe, which can be used to control the GrowBot.

In order to monitor the status of the GrowBot, the user can interact with the GrowBot through the touchscreen display and through the remote webserver interface. On the display, the user can get simple information on the current value of the sensors: temperature, humidity, CO₂, and water level. This might be information, which is good to easily be able to see when physically present next to the GrowBot, so it is shown on the front page of the display on the GrowBot. On the webserver interface, the user can get more extended information including the current and historical data of the sensors in graph form, photos from the RGB camera, and information on when actuation commands were performed

on the GrowBot. In the graphs, the user can select to see data from the last hour, the last day, the last week, or from a user-defined time interval. The users can also toggle between the photos taken by the RGB camera each 8 h (and possibly make their own still motion video of the growth happening in their GrowBot, and program the camera to take photos at other time intervals). Additionally, on the webserver, the users can export all the data in CSV-format for further analysis in their preferred software such as Microsoft Excel. Figure 10 shows an example of part of the webserver interface for monitoring the sensor data. Here, the temperature and humidity data for one day (24 h) is shown, and the representation in graph form makes it easy to recognize how the temperature drops from 11 p.m. until 5 a.m., and the humidity rises in the same interval. In this case, this corresponds with actuator commands to turn off the lights at 11pm and turn on the lights at 5 a.m.

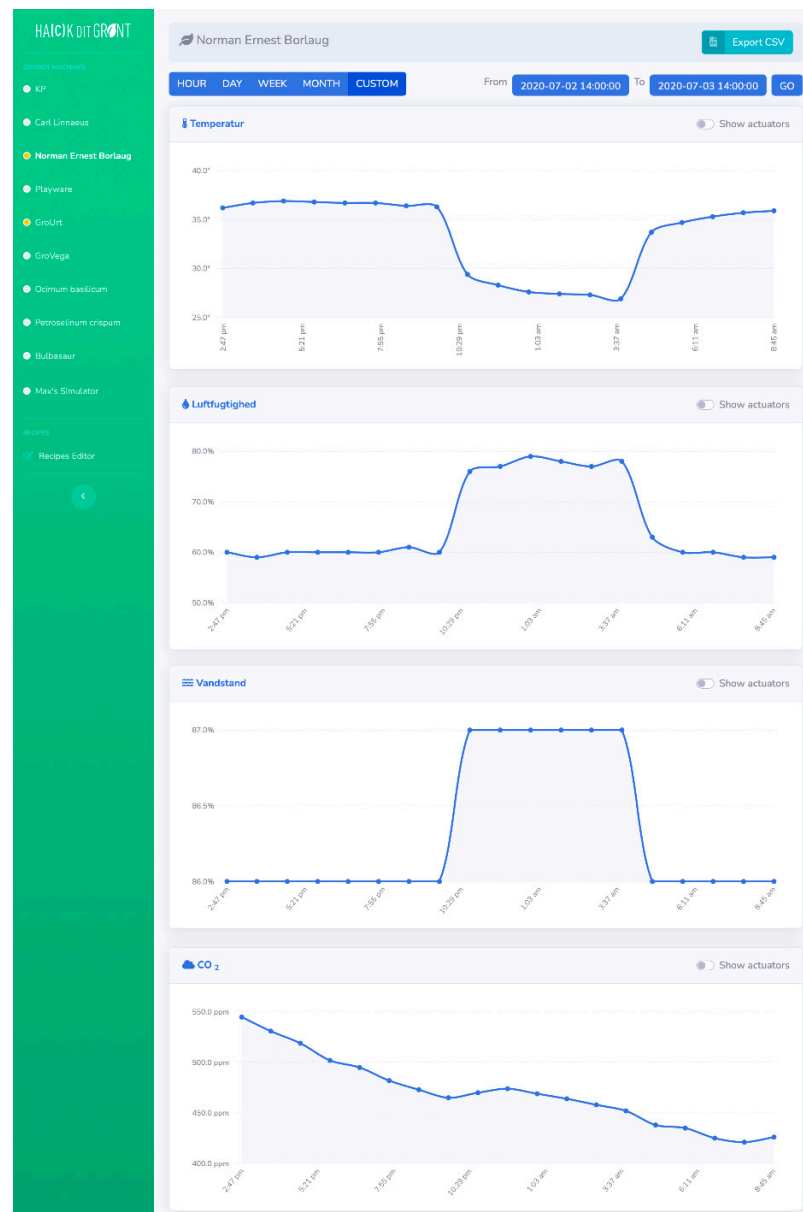


Figure 10. The web interface shows the status of the GrowBot sensors. The graphs show temperature, air humidity, water level, CO₂ (top to bottom).

6. Application and Domain Adoption

The website for the GrowBots is built to provide ample information for the teachers to use the GrowBots to match the curriculum and learning objectives for the school classes. Apart from containing a detailed teachers' guide, it provides 16 different resources for teaching themes. Most importantly, the website also contains a number of cases (initially 4 cases), which gives appealing themes to structure the teaching during a period of time while using the GrowBots (see Figure 11). The cases include "Grow green and become an urban farmer", "Grow green and serve vegetables", "Grow green and eat green proteins", and "Grow green with limited resources". Each case contains 15–20 phases for the school class to go through. These can include work on the UN goals, making a logbook, obtaining knowledge on plants and growth, experimenting with healthy food, the engineering working process, creating hypotheses, programming, data analysis, making evaluation, etc. Thereby, the teaching process for using the GrowBots is facilitated for the teacher, who can use one of the cases and pick all the phases or a sub-part of these phases to have a ready-made process for the teaching in the class over a certain teaching period, ensuring that the class works according to the curriculum and certain learning objectives. With this, we aim for the teacher to have a *plug-and-play* solution both in the form of the GrowBot tool itself, and in the form of the teaching material and process.

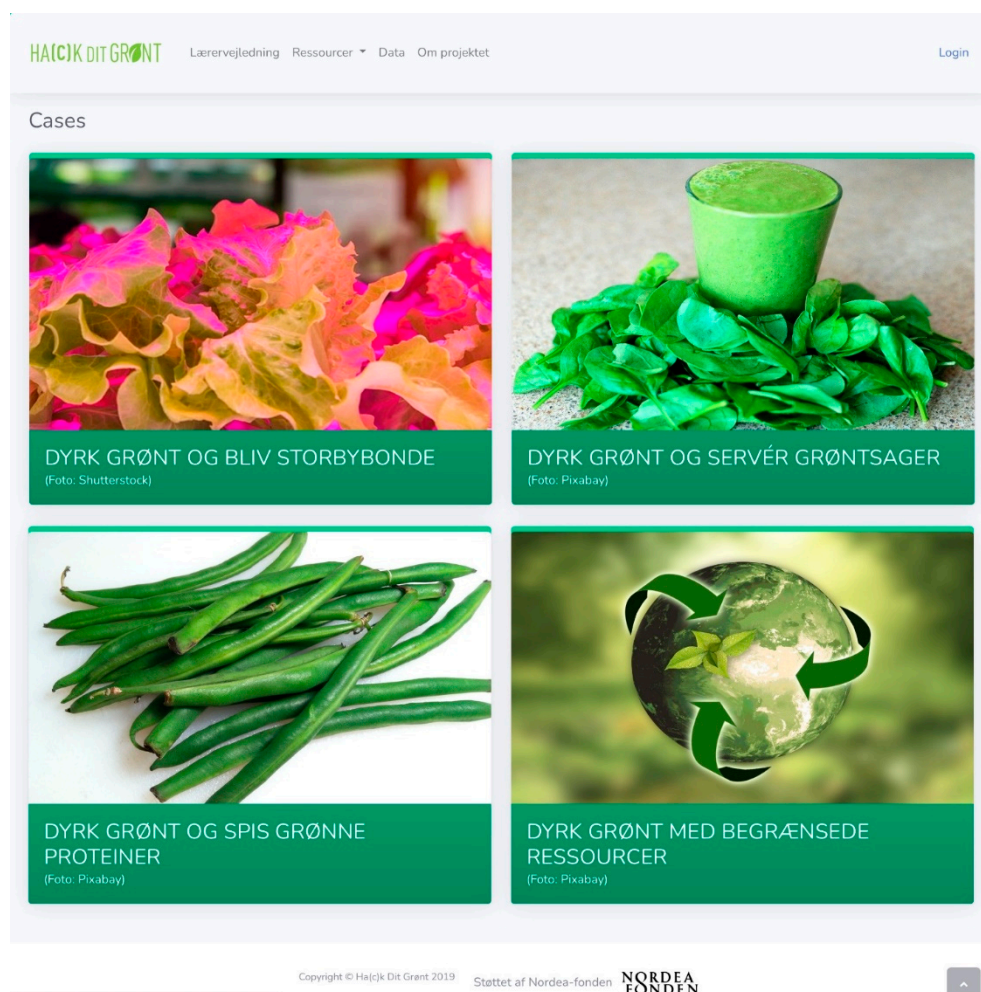


Figure 11. On the web-page (in Danish), users can easily select a case of interest. Clockwise from upper left corner, the cases are translated to "Grow green and become an urban farmer", "Grow green and serve vegetables", "Grow green with limited resources", and "Grow green and eat green proteins". Under each case, there are 15–20 educational tasks related to the theme of that case.

With the extensive work to provide cases and ample teaching material, we place the developed technology in a context where it is easy to become adopted by the potential users. We do not only make the new technology tool itself user-friendly, but we make the whole ecosystem around the technology user-friendly. This allows the operators and users (in this case, the teachers and pupils) to easily adopt the new technology in their daily life (in this case, in their school life).

Figure 12 illustrates the design of the teaching intervention for one of the cases with the full process of 18 smaller educational tasks. The first 3 weeks consist of framing the learning tasks, introduction to the work process and methodology, smaller programming tasks, etc. The following 5 weeks consist of experimentation with growth in the GrowBots, and field work related to vegetables. Finally, the process ends with presentation and evaluation. For each of the 18 smaller educational tasks, there is ready-made material for teacher and pupils in order to facilitate the domain adoption. The comprehensive material helps the teacher to easily get started and to feel confident to reach the teaching curriculum goals. According to our methodology for comprehensive system design, this work on domain adoption is crucial to ensure deployment in society of the developed technology.

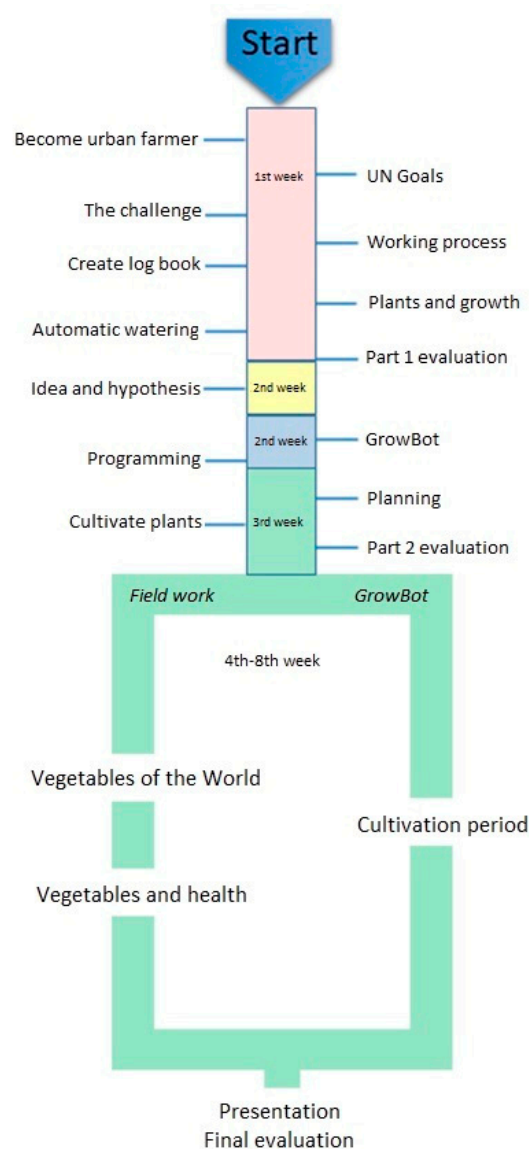


Figure 12. The educational process as it is presented to the teacher with 18 different educational tasks.

7. Experiments and Results

Initially, as the basis for future educational implementations, a number of fundamental experiments were performed to verify the controllability of the growth conditions in the GrowBot. Here we will look at the example of controlling the temperature. One such experiment would turn on the white light for 2 h followed by turning on white, blue, and red light for one hour. The fan and air change (1000 mL/h) would be turned ON when the temperature would exceed 35 °C. Figure 13 shows the result of this experiment. The initial temperature was 26.0 °C, and in the 2 h period of white light condition, the temperature rose and stayed at approximately 31 °C, ending at a maximum temperature of 31.4 °C. At this point, the white, blue, and red lights were all turned on, and the temperature rose to a maximum of 35.2 °C. The fan and air change turned on, and the temperature fell slightly to stay around 34.8 °C. When all lights were turned off after 1 h, the temperature again fell to the initial 26.0 °C over a time period of 1.5 h.

Temperature variation in the growth chamber due to white light and light mixture ($T_c = 35^\circ\text{C}$)

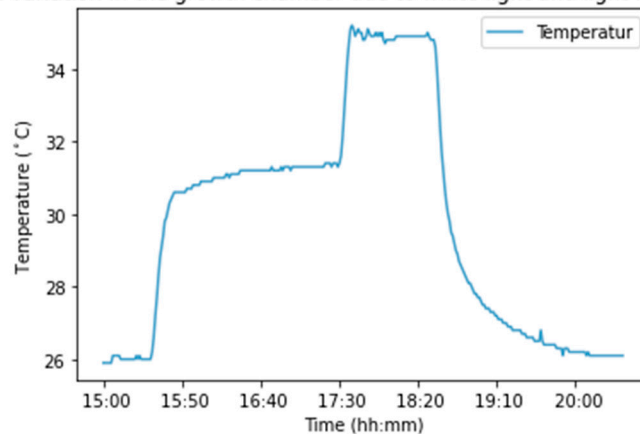


Figure 13. The temperature in a Growbot experiment when white light is turned on at 15:30, white, blue, and red light turned on at 17:30, and all lights turned off at 18:30.

To test the robustness of the developed GrowBots further, and to gain insight into the growth process inside the GrowBots to inform the educational material development, we performed an experiment in which we planted seeds in nine GrowBots controlled with different parameter settings (programs) for 43 days. The seeds were sown in small rockwool cubes placed in cups that were put into the holes in the water tray. Each GrowBot can contain 16 such cups. The experiment consisted of 4 rows of 4 cups with seeds for parsley, basil, red lettuce, and green lettuce. In this experiment, there was no germination prior to sowing the seeds in the cups.

The experiment was performed with nine GrowBots (see Figure 14). All GrowBots had a day light cycle of 18 h of light and 6 h of no light during each 24 h period. Three GrowBots had red light (730 nm, set to 40% of max intensity) and a small component of white (full spectrum) light (set to 16% of max intensity), while the other six GrowBots had blue light (458 nm, set to 40%) and a small component of white (full spectrum) light (set to 16%). The three GrowBots with red light and three of the GrowBots with blue light had nutrient added to the water basin (10 g for 9 L water) at the beginning of the experiment, while the last three GrowBots had (i) nutrient added from the nutrient bottle at the beginning (10 g from the 200 mL bottle), (ii) nutrient added in small drops from the nutrient bottle with 10 mL/6 h (10 g in total in the 200 mL bottle), and (iii) no nutrient. In all cases, the aeration and air circulation fans were set to the same low level.



Figure 14. Seven of the nine GrowBots used in the lab for the experiment.

The reason to experiment with different color in the GrowBots is that the color of light will affect the growth of the plants. Therefore, the controllability of this parameter can be used as a learning lesson for the pupils. The effect of blue light on plants is related to chlorophyll production, and blue light promotes root development and strong, stocky plant growth. Red light is responsible for making plants flower and produce fruit, and therefore works best in the flowering phase. Hence, for the initial growth of the food plants in the GrowBots, a teacher and/or pupils could possibly make the hypothesis that they would observe stronger and higher initial growth under blue light than under red light conditions.

The experiment was performed over 43 days in the period 30 June 2020–12 August 2020. During this period, all the GrowBots were running automatically with the control program (program), and water was added 2–3 times to the water tray of each GrowBot to keep the water level high enough in the tray.

During the experiment, the sensor data from all nine GrowBots could be monitored in real-time on the web-interface of the server, and further, the data was stored on the server for analysis of the full dataset. Figure 10 shows the web-interface visualization of the sensor data for one of the GrowBots. Here it can be observed how the temperature, humidity and CO₂ level are dependent variables, and how they consistently changed based on the control of the light color and level.

As can be seen with the example in Figure 15, as a result of the experiment, it was found that clear visible growth of the plants only happens after a few weeks of growing of the seeds. This informed the educational material to include a recommendation for the teacher (and pupils) to perform germination and seed growing in cups for 2–3 weeks prior to initiating the growth in the GrowBot. In this way, the teacher and pupils can select the cups with successful seed growth to be put into the GrowBot to ensure an interesting growth development and comparison between programs to be performed by the pupils.

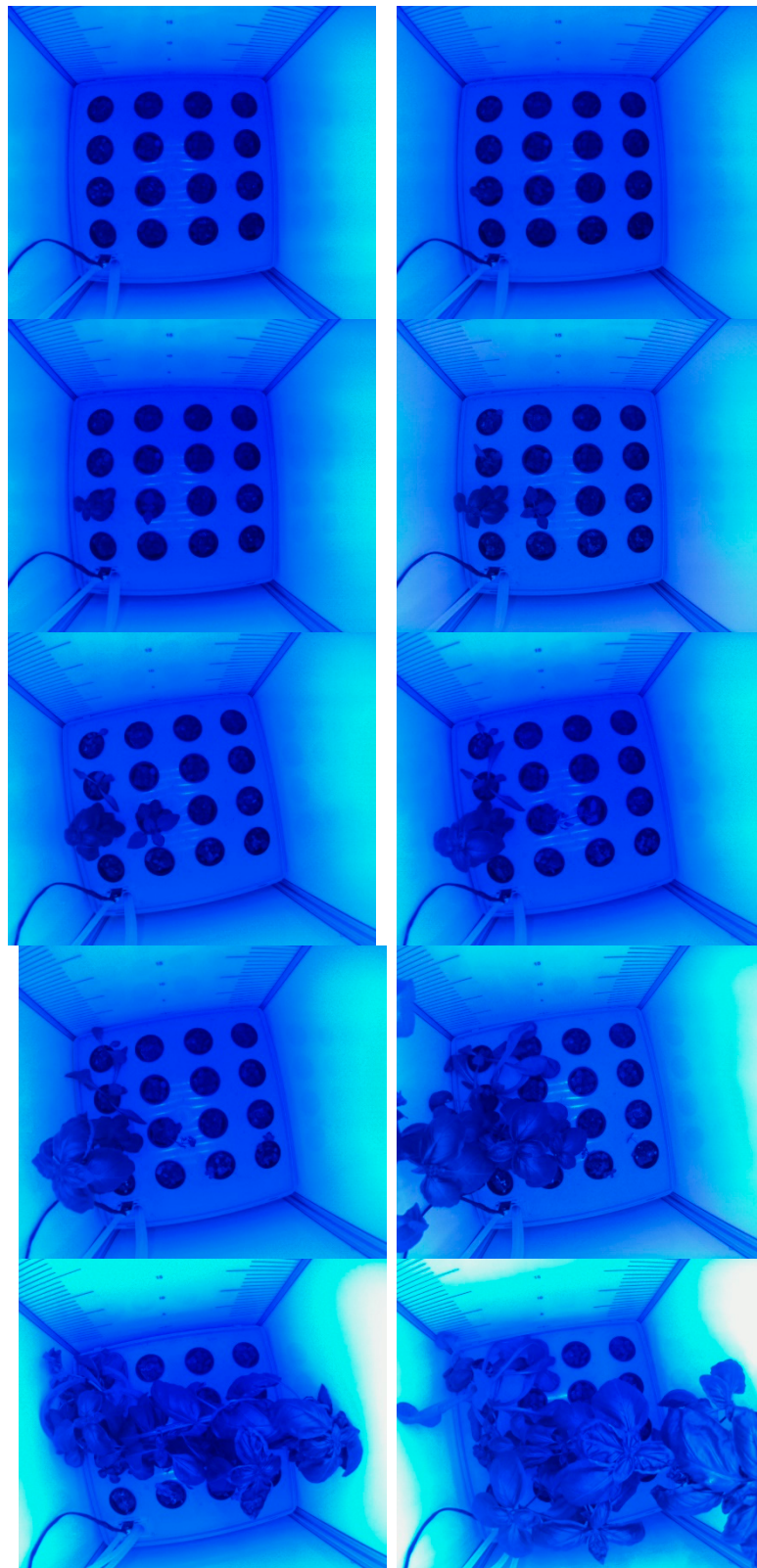


Figure 15. Photos from the camera mounted inside one of the GrowBots showing the growth of a basil plant in a GrowBot during 43 days (images from the last 35 days).

At the end of the experiment after 43 days, all plants were harvested, including careful separation of the roots from the rockwool (see Figure 16 for examples). Afterwards, the roots were carefully dissected from the plant, and the rest of the plant was used for

further measurement of height, weight, and leaf width. The results of the experiment are summarized in Table 4. All the plants grown in the GrowBots with blue light had a mean height of 4.3 cm at the time of harvest at the end of the experiment, while the plants in the GrowBots with the red light had a mean height of 1.0 cm. The mean weight of the plants without roots from GrowBots with blue light was 1.31 g, and from GrowBots with red light the mean weight was 0.05 g. The mean width of the three largest leaves from the GrowBots with blue light was 1.13 cm, whereas it was 0.19 cm for the leaves from the GrowBots with red light. The statistical analysis with the unpaired t-test showed that the difference in plant height and leaf width between the blue light and red light conditions is statistically significant at $p < 0.01$ (see Table 4).



Figure 16. (Top): The tray with growth of green lettuce, red lettuce, basil, and parsley after 43 days in a GrowBot. Bottom: Harvest of one row of plants grown in two different GrowBots with each program during the 43 day experiments, including their roots (left): basil (from the 2nd row of a GrowBot named Carl), (right): parsley (from the 1st row of a GrowBot named KP).

Table 4. The mean plant height, plant weight, and leaf width from the GrowBots with blue and red light. SD for standard deviation. * indicates statistical significance, $p < 0.01$.

	Height (Mean)	SD	Weight (Mean)	SD	Leaf Width (Mean)	SD
Blue light	4.3 cm	7.8	1.31 g	5.87	1.13 cm	1.83
Red light	1.0 cm	2.2	0.05 g	0.21	0.19 cm	0.49
<i>p</i> -value	0.006 *		0.1418		0.0007 *	

Even if there was no germination and seed growing in cups for 2–3 weeks prior to initiating the growth in the GrowBots, it is possible to observe the difference between growth of food plants in blue and red light conditions. Hence, the experiment would confirm the hypothesis for the teacher and pupils that initial growth under blue light would produce stronger and higher food plants than under red light conditions. A number of seeds did not germinate, as can be seen on Figure 15, and this may have influenced the results of the experiments, so a germination process may be advisable for the pupils before starting the growth experiments, in order to have a clear basis for comparison and testing their hypotheses.

The results verified that the GrowBot technology can work as a controllable environment for some food plant growth, which was the main aim as the foundation for a novel educational robotic system. To verify the suitability and robustness of the GrowBots in school use, we performed a pilot test with 6 GrowBots in the school Tinderhøj Skole, in Denmark. The 6 GrowBots were given to a 7th grade class (13–14 years old pupils) as part of a teaching period in which they used the case “Grow green and become an urban farmer” from the teaching material. As illustrated in Figure 12, the school class went through the full process of 18 smaller educational tasks. The first 3 weeks consisted of framing the learning tasks, introduction to the work process and methodology, smaller programming tasks, etc. The following 5 weeks consisted of experimentation with growth in the GrowBots, and field work related to vegetables, and the process ended with presentation and evaluation.

During the 8 weeks of working with the case “Grow green and become an urban farmer”, the school class was able to go through the educational tasks. The pupils were divided into groups, with each group responsible for a GrowBot. The 6 GrowBots worked as intended during the intervention period, and the pupils were able to grow from seeds in the GrowBots. They were able to analyze the data remotely on the server and compare results based on the different programs programmed by the different groups. Figure 17 shows a snapshot from the intervention in the school with groups of school pupils programming their GrowBots, installing the micro:bit on the GrowBot.



Figure 17. Two groups of the school pupils interacting with the GrowBot during the 8 weeks intervention at the school Tinderhøj Skole, in Denmark.

8. Discussion and Conclusions

We developed the GrowBot as a robotic system for growing food, allowing environmental growth parameters to be expressed and controlled in a program. In this way, food plant growth can be automatically controlled in the GrowBots, allowing the programs to be developed by different means such as machine learning, exchange, or user programming. Here, we focused on the user programming in the educational setting based on an initial experimentation to show the appropriateness and robustness of the GrowBots. The experiment verified that we can control the different parameters, and that this can control the growth of the food plants. In this case, the hypothesis that control to make an environmental condition with blue light would result in higher and larger plants than red light could be verified with the experiment controlling those conditions in the GrowBots. In this case, we are able to set the parameters to control the growth of natural food plants. Further, the experimentation in school settings showed that the comprehensive system design method resulted in a deployable system well adopted in the educational domain. Indeed, it indicated how easy it was to adopt the new approach and technology in the school when all material, from the lower layers to the higher layers, was designed, developed, and tested according to the comprehensive system design method. The important lessons learnt are that (i) lower layers must be fully functional, well debugged, and robust in order for any higher layers not to fail, and (ii) higher layers are crucial to be incorporated in the design and development process if the lower layer technology is ever to become adopted in society use.

We have reported how the GrowBot can be programmed for optimizing the food plant growth and allow remote monitoring of the growth and the environmental conditions in the GrowBot, functioning as a robotic system. This constitutes the foundation for an educational robotic system for use by school pupils. As an educational tool, the GrowBot allows school classes hands-on experiences with programming for controlling the smart farming environment of the GrowBot. By exploring and programming the control parameters of this robotic system, the students are exploring and controlling different growth conditions for the food plants. The educational impact of this will be investigated in our future work, which includes a school intervention with more than 500 school pupils.

Author Contributions: Conceptualization, H.H.L., M.E., G.R.-H. and R.A.; Investigation, H.H.L., G.R.-H. and R.A.; Methodology, H.H.L., G.R.-H. and R.A.; Data curation, M.O. and R.A.; Funding acquisition, R.A.; Project administration, R.A.; Software, N.E.J., M.L., M.v.S. and A.V.; Writing—original draft, H.H.L.; Writing—review & editing, H.H.L., M.E., M.O., G.R.-H. and R.A. All authors have read and agreed to the published version of the manuscript.

Funding: The project is funded by the Nordea-fonden grant 02-2018-1485.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Acknowledgments: The authors would like to thank the participating teachers and pupils at the schools, Tinderhøj Skole and Grønnemose Skole. Also, the authors would like to thank Line Andresen, Poul Thorbjørn Sørensen, and Jacob Svagin at DTU Skylab for their valuable contribution to the development of the Growbots, and Rishav Bose for experimental work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gurung, C.T.; Bhandari, J.B.; Gurung, A. Evaluation of hydroponic cultivation techniques as a supplement to conventional methods of farming. *J. Agric. Technol.* **2019**, *6*, 56–57.
2. Despommier, D. The vertical farm: Controlled environment agriculture carried out in tall buildings would create greater food safety and security for large urban populations. *J. Verbrauch. Lebensm.* **2011**, *6*, 233–236. [[CrossRef](#)]
3. Harper, C.; Siller, M. OpenAG: A globally distributed network of food computing. *IEEE Pervasive Comput.* **2015**, *14*, 24–27. [[CrossRef](#)]
4. Ferrer, E.C.; Rye, J.; Brander, G.; Savas, T.; Chambers, D.; England, H.; Harper, C. Personal Food Computer: A new device for controlled-environment agriculture. In Proceedings of the Future Technologies Conference, Vancouver, BC, Canada, 15–16 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1077–1096.

5. Goldstein, H. MIT Media Lab Scientist Used Syrian Refugees to Tout Food Computers That Didn't Work. *IEEE Spectrum*. **2020**. Available online: <https://spectrum.ieee.org/mit-media-lab-scientist-used-syrian-refugees-to-tout-food-computers> (accessed on 27 May 2022).
6. Van, L.-D.; Lin, Y.-B.; Wu, T.-H.; Lin, Y.-W.; Peng, S.-R.; Kao, L.-H.; Chang, C.-H. PlantTalk: A Smartphone-Based Intelligent Hydroponic Plant Box. *Sensors* **2019**, *19*, 1763. [[CrossRef](#)] [[PubMed](#)]
7. Mehra, M.; Saxena, S.; Sankaranarayanan, S.; Tom, R.J.; Veeramanikandan, M. IoT based hydroponics system using Deep Neural Networks. *Comput. Electron. Agric.* **2018**, *155*, 473–486. [[CrossRef](#)]
8. Marques, G.; Aleixo, D.; Pitarma, R. Enhanced Hydroponic Agriculture Environmental Monitoring: An Internet of Things Approach. In *Computational Science—ICCS 2019*; Rodrigues, J.M.F., Cardoso, P.J.S., Monteiro, J., Lam, R., Krzhizhanovskaya, V.V., Lees, M.H., Dongarra, J.J., Sloot, P.M.A., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11538.
9. Angeloni, S.; Pontetti, G. RobotFarm: A Smart and Sustainable Hydroponic Appliance for Meeting Individual and Collective Needs. In *Innovative Mobile and Internet Services in Ubiquitous Computing, Proceedings of the Advances in Intelligent Systems and Computing, IMIS 2019, Sydney, Australia, 3–5 July 2019*; Barolli, L., Xhafa, F., Hussain, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 994.
10. Stevens, J.D.; Shaikh, T. MicroCEA: Developing a Personal Urban Smart Farming Device. In Proceedings of the 2nd International Conference on Smart Grid and Smart Cities, Kuala Lumpur, Malaysia, 12–14 August 2018; IEEE Press: Piscataway, NJ, USA, 2018.
11. Boanos, A.; Chandrababu, S.; Bastola, D.R. Automation of Personal Food Computers for Research in Drug Development and Biomedicine. In Proceedings of the 7th International Congress on Advanced Applied Informatics, IIAI-AAI, Yonago, Japan, 8–13 July 2018; pp. 655–658.
12. Lund, H.H.; Pagliarini, L. RoboCup Jr. with LEGO MINDSTORMS. In *Proceedings 2000 ICRA. Millennium Conference, Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24–28 April 2000, Symposia Proceedings (Cat. No.00CH37065)*; IEEE Press: Piscataway, NJ, USA, 2000; Volume 1, pp. 813–819.
13. Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; et al. Scratch: Programming for all. *Commun. ACM* **2009**, *52*, 60–67. [[CrossRef](#)]
14. Fraser, N. Ten things we've learned from Blockly. In Proceedings of the 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond), Atlanta, GA, USA, 22 October 2015; pp. 49–50.
15. Pacheco, M.; Fogh, R.; Lund, H.H.; Christensen, D.J. Fable II: Design of a modular robot for creative learning. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 6134–6139.
16. Pacheco, M.; Moghadam, M.; Magnússon, A.; Silverman, B.; Lund, H.H.; Christensen, D.J. Fable: Design of a modular robotic playware platform. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 544–550.
17. Microbit. 2020. Available online: www.microbit.org (accessed on 7 April 2022).
18. Lund, H.H.; Klitbo, T.; Jessen, C. Playware technology for physically activating play. *Artif. Life Robot.* **2005**, *9*, 165–174. [[CrossRef](#)]
19. Lund, H.H. Playware Research—Methodological Considerations. *J. Robot. Netw. Artif. Life* **2014**, *1*, 23–27. [[CrossRef](#)]
20. Brooks, R. A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.* **1986**, *2*, 14–23. [[CrossRef](#)]
21. Arkin, R.C. *Behavior-Based Robotics*; MIT Press: Cambridge, MA, USA, 1998.