Planty

TechiStudio

Table of contents

Who are we?

We are seventh grade students of Érdi Vörösmarty Mihály Gimnázium. Our names are Andorfer Zalán, Botyánszki Zétény, and Hanuszka Zente. We have been learning robotics for 2 years, our first project was a school presentation. For that we completed a dancing and calculating robot, and a searching algorithm. Since then, as a part of the workshop we regularly meet and work together. We have a dedicated classroom where we not only prepare for competitions but do fun projects and experiment. Six of our classmates are also part of the workshop, some of them are more experienced, while some only started this year. We help them learn programming and design principles regularly. Last year we participated in the RoboMisson category, where we qualified for the national finals, and in this December, we participated in the EduCup.

We share our creations and achievements for those interested on our Youtube channel [TechiStudio]() and our website https://techistudio.github.io/

Because this project consists of many components, we tried to group them, and disperse them among ourselves, so we can keep track of everything. Zalán took part in the programming and research, Zétény in design building and programming, while Zente dealt with planning and building the excavator. We work together planning and writing documentation for the robot.

Introduction

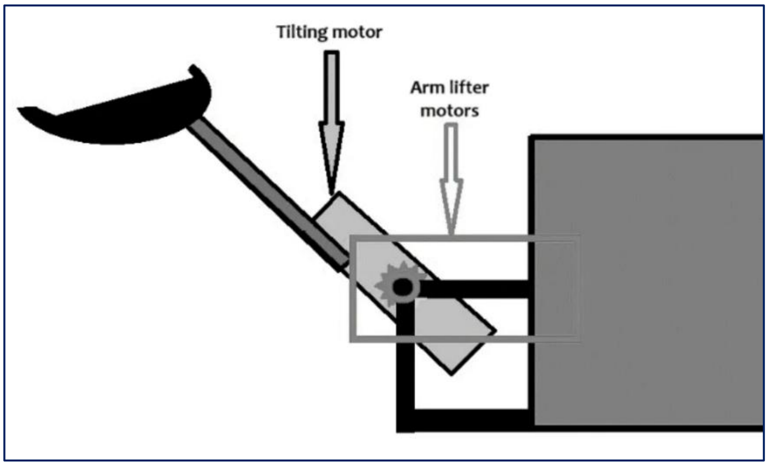
The goal of our project is making food easy to grow on adequate planets while exploring space and on densely populated or inhospitable areas on earth. It delivers soil or regolith using an automatic excavator robot to a measuring station, that examines the soil to decide what to plant. It also waters and plants and sorts the chosen crop, while giving useful analytical data to the plant's caregivers. The measurement is based on NPK values which are compared to the preferences to the available crops to decide what to plant. The process prepares the plants to vertical farming, a technology already showing promise for the exact same problem we're trying to solve. However, if the soil is not adequate for growing plants, the mechanism can still provide information about the properties of the given soil and can help map the composition of the planet's surface, advancing human understanding about these celestial bodies.

The implementation of the project

As we mentioned before the project consists of two parts: the base and the excavator. The base uses two microcontrollers. First it has a LEGO Spike, because of its ease of use and the fact, that we already had Spike compatible parts and experience with them from previous competitions. Unfortunately, the NPK sensor and the pump couldn't be controlled with Spike, so we must use another microcontroller - in our case an Arduino Leonardo. The excavator also uses a Lego Spike.

Problems and evolution

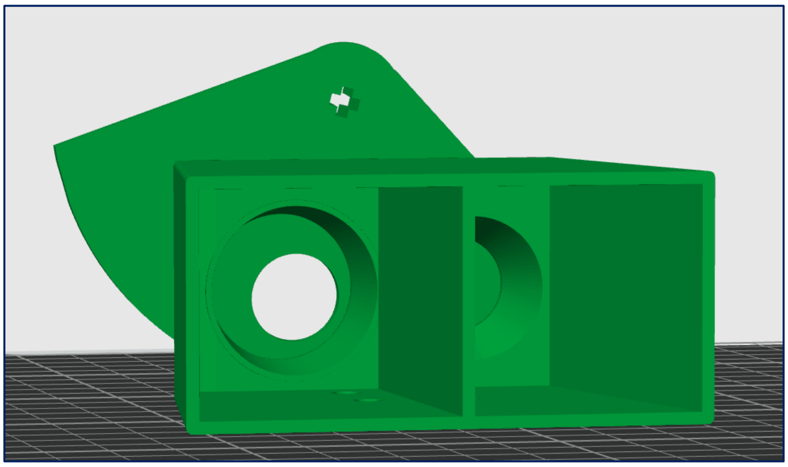
Lifting and tilting of the collecting robot's excavator bucket: For tilting, a motor needs to be lifted alongside the bucket filled with soil. This is difficult even for large motors. Therefore, the robot uses a smaller excavator bucket and fills it up multiple times. Furthermore, the arm's anchorage is not at the end but near the centre of mass, meaning that the tilting motor does not need to be lifted.



The mass distribution of the excavator robot: A full bucket makes the robot very front-heavy, which we counterbalanced with metal weights mounted on the back of the robot.

The movement of the excavator robot on uneven terrain: Smooth rubber wheels produced a lot of wheelspin on the uneven ground, and we must keep the motors and sensors away from the ground to protect them from soil particles. Therefore, the robot is equipped with tracks, and the gears driving the tracks are arranged in a triangular formation to keep the system as far from the ground as possible.

Communication between the two control units of the base: Initially, we intended to solve the communication between the Arduino and Spike by transmitting the data using an RGB LED and a colour sensor, but this did not work because the colour sensor looks at the reflected colour, not the emitted one. After that, we came up with the idea that the Spike should count the Hz of the LED's blinking, which is emitted by the Arduino. This was also changed to count the time the light is on; this made programming easier.



Seed dosing: We wanted to dose several types of seeds using as little electronics as possible. To achieve this, we designed a special container that can dose two types of seeds with only one motor.

Storage of the containers: The containers need to be moved to different locations for planting and then for germination. We use a conveyor belt for this, which is controlled by the LEGO Spike motors push the containers in and control the belt. We solved the scheduling of the process with the help of the timer parameter.

The NPK sensor: Plants do not thrive in every soil; they prefer certain soil compositions. During our research, we concluded that the essential nutrients for plant development are nitrogen, phosphorus, and potassium. The NPK sensor measures their quantities and then differentiates based on these results.

The excavator robot

This robot has five motors, two for travelling and three for moving the arm (one to tilt the excavator bucket and two to lift the bucket and the motor). The robots control panel is a LEGO Spike hub. The robot's job is to go to a location and collect samples from the soil. First it goes to a specified location then picks up samples with its excavator bucket. After that it determines its location using its distance sensor and two markers. Then takes the samples to the base where the sample is emptied from the excavator bucket into the funnel for analysis.

Navigating the collector robot to the base: We place two markers in the terrain. The robot determines its distance from these using a distance sensor. From the two distances, it calculates its distance and angle from the base using the following formulas. (a, b the two distances, x, y the positions in the coordinate system.)

$$x = \frac{b^2 + field\_size^2 - a^2}{2 \cdot field\_size} \quad y = \sqrt{b^2 - x^2}$$

$$distance(base, robot) = \sqrt{(x - base\_origin)^2 + (field\_size - y)^2}$$

$$angle(base, robot) = tan^{-1}\left(\frac{x - base\_origin}{field\_size - y}\right)$$

What is NPK?

NPK is the abbreviation for the three most important plant nutrients: nitrogen (N), phosphorus (P), and potassium (K). These are essential for the healthy development of plants. Nitrogen mainly aids the growth of leaves and green mass, making it particularly important during the vegetative stage. Phosphorus primarily plays a role in root development, flowering, and fruit formation. Potassium regulates water balance, strengthens plant cells, and enhances resistance to diseases and extreme weather conditions. Plants require different NPK ratios at various stages of development: initially, more nitrogen is needed, while phosphorus and potassium become more prominent during flowering and fruit ripening. It is important to know the needs of our plants and to choose the appropriate nutrient composition for the robot accordingly. Therefore, NPK is one of the cornerstones of plant cultivation, which must be applied consciously for sustainable and efficient farming.

The base

The Arduino

The workings of the devices controlled by the Arduino are more complex, so below we go into detail on each component. If needed a wiring diagram is available on the 14th page that you can refer to.

The NPK sensor

We use a JXBS-3001-NPK-RS model NPK sensor from JXCT. It needs 12-24 volts of power, which we achieve by linking two 9-volt batteries together. These sensors use ion-sensitive electrodes to determine the amount of Na, Ph, K ions in the soil. It communicates with the RS-485 protocol using a MAX485 converter. To send and receive data it uses the A and B wires which the converter translates for us and transmits to and from the RO (Receiver Output) and the DI (Driver Input) pins. In this example they are defined under the mod object. First, we signal we want to write data with DE (Driver Enable. Must be high, if we want to write data) and RE (Receiver Enable. Must be low, if we want to read data) and write the inquiry:

```
digitalWrite(DE, HIGH);
digitalWrite(RE, HIGH);
delay(10);
mod.write(inquiry_data, sizeof(inquiry_data));
mod.flush();
```

Then we enable reading and wait for the response. The loop has a timeout to stop waiting for data after an unreasonable amount of time:

```
digitalWrite(DE, LOW);
digitalWrite(RE, LOW);
startTime = millis();
while ( millis() - startTime <= TIMEOUT ) {
  if (mod.available() && byteCount<sizeof(values) ) {
    values[byteCount++] = mod.read();
  }
}
```

Then we return the 5th byte of data as that carries the values for the measurement:

```
return values[4];
```

The inquiry codes are the following:

```
const byte nitro[] = {0x01, 0x03, 0x00, 0x1e, 0x00, 0x01, 0xe4, 0x0c};
const byte phos[] = {0x01, 0x03, 0x00, 0x1f, 0x00, 0x01, 0xb5, 0xcc};
const byte pota[] = {0x01, 0x03, 0x00, 0x20, 0x00, 0x01, 0x85, 0xc0};
```

The pump

We use 3-6v mini water pump to water the plants. Controlling this is straightforward, although you must use a relay with a battery to provide enough current. All you must do is give power to the relay:

```
digitalWrite(PUMP_PIN, HIGH);
delay(pump_time);
digitalWrite(PUMP_PIN, LOW);
```

The distance sensor

We use the HC-SR04 ultrasonic distance sensor. We can interface with the sensor with digital pins. The sensor has two: trigPin and echoPin First we ask for a measurement by switching the trigPin to high:

```
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

Then to calculate the distance we need to multiply the answer time with 0.017:

```
duration = pulseIn(echoPin, HIGH);
old = distance;
distance = duration * 0.017;
```

Communication

We communicate between Spike and the Arduino with LEDs and a colour sensor. We communicate numbers each of which has a meaning:

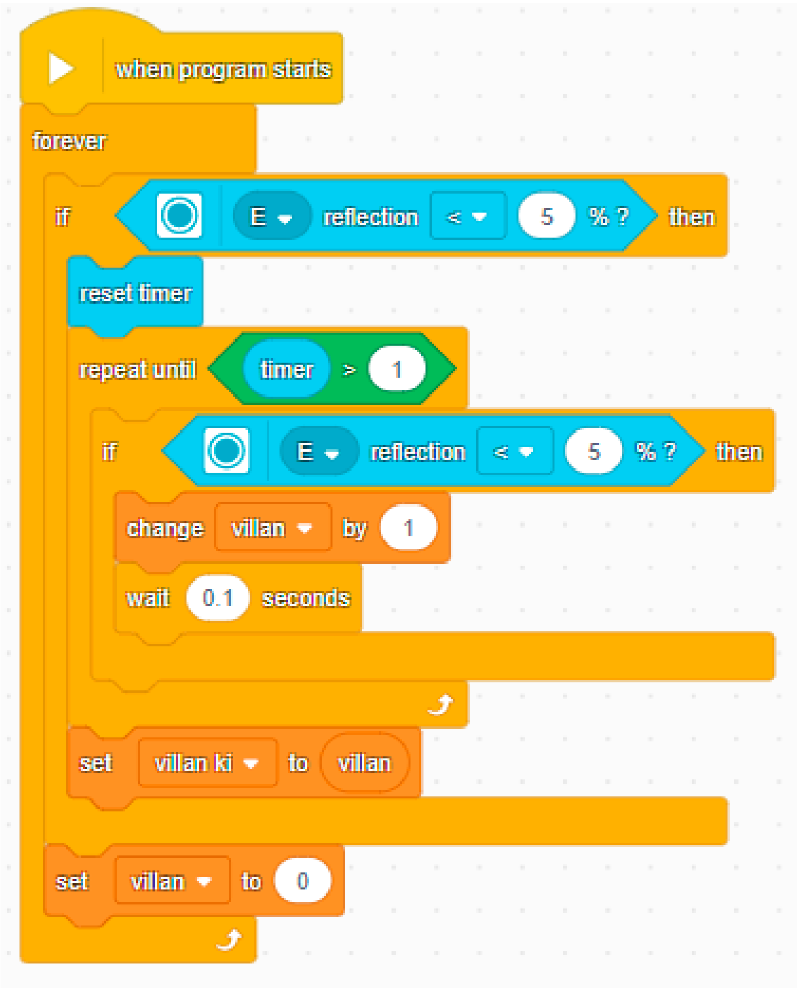
Number	Information
1	Soil was delivered, start the program
2	Plant wheat
4	Plant beans
8	The soil is of unusable quality, throw it out.

Each of the numbers relates to how long the LED is on. We get the length by multiplying the signal by 100 ms. The codes communicating what crop to plant are further apart to avoid misdetection.

To send messages we use an analog pin with an RGB LED:

```
void ledSignal(int code) {
  analogWrite(R, 255);
  delay(code * 100);
  analogWrite(R, 0);
}
```

On the Spike if we detect that the light is on, we count how long it shines by checking every 100ms and incrementing a variable:



The program logic

To demonstrate the flow of the program below it is written in pseudocode to shorten it. The controlling of the components was already demonstrated, so their programming is absent from this snippet.

```
loop begin
  if enough soil has been delivered
    send word to the Spike
    water the soil
    while we havent measaured the same value five times
      measaure the npk
      signal it to the dashboard
    decide on the crop to plant
    signal it to the dashboard
    signal it to the spike
end
```

Comparing the NPK values

To decide what to plant we must compare the measured NPK values with the preset "perfect" values for the given plants:

Plant	N	P	K
wheat	25	12	18
beans	10	40	20

Please note that these are just ratios; beans don't need nearly four times the phosphorus of wheat.

There are multiple different ways to compare the NPK values each with their own upsides and downsides. We compare them by calculating the difference between all NPK values and the average, then comparing those. For example, if N is 5 and the sum of N, P, and K is 21, then the average is 7, so one value is 7/21 of the whole, while N is 5/21 of the whole, so the difference 2/21. Let's propose that on a different NPK value we go through the same process and get 7/32. Then the difference between those two values becomes |7/32 - 2/21| = around 0.13. We do this for all N, P, and K values and then compare the difference between the wheat and the measured value and of the beans and measured value like so:

```
int getPlant(byte N, byte P, byte K) {

  double npk[3] = {N, P, K};
  double wheat[3] = {25, 12, 18};
  double beans[3] = {10, 40, 20};
  double wheatDiff = abs((1 / 3 - npk[0] / (N + P + K)) -
                     (1 / 3 - wheat[0] / (wheat[0] + wheat[1] + wheat[2]))) +
                     abs((1 / 3 - npk[1] / (N + P + K)) -
                     (1 / 3 - wheat[1] / (wheat[0] + wheat[1] + wheat[2]))) +
                     abs((1 / 3 - npk[2] / (N + P + K)) -
                     (1 / 3 - wheat[0] / (wheat[0] + wheat[1] + wheat[2])));
  double beansDiff = abs((1 / 3 - npk[0] / (N + P + K)) -
                     (1 / 3 - beans[0] / (beans[0] + beans[1] + beans[2]))) +
                     abs((1 / 3 - npk[1] / (N + P + K)) -
                     (1 / 3 - beans[1] / (beans[0] + beans[1] + beans[2]))) +
                     abs((1 / 3 - npk[2] / (N + P + K)) -
                     (1 / 3 - beans[0] / (beans[0] + beans[1] + beans[2])));

  if (npk[0] < 10 or npk[1] < 10 or npk[2] < 10) {
    Keyboard.write(KEY_F16);
    delay(10);
    Keyboard.releaseAll();
    return 3;
  } else if (wheatDiff < beansDiff) {
    Keyboard.write(KEY_F17);
    delay(10);
    Keyboard.releaseAll();
    return 1;
  } else {
    Keyboard.write(KEY_F18);
    return 2;
  }
}
```

The dashboard

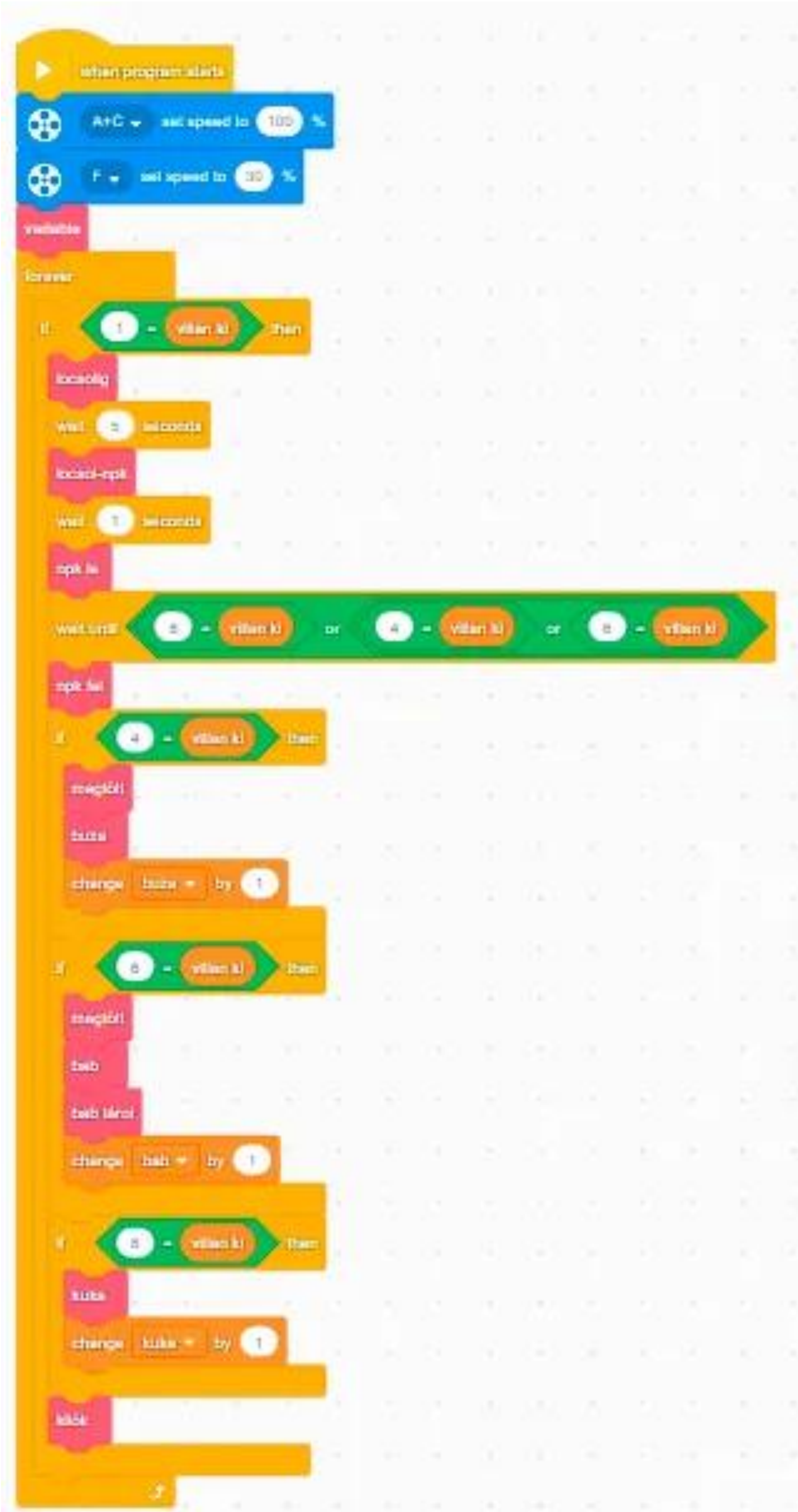
To monitor and record the values measured by the robot we created a custom dashboard. It communicates with the Leonardo with keypresses of the F keys not available on a modern keyboard. First the Arduino presses an F key corresponding to the given value, then writes the number, and closes the communication with another F key press, as seen below:

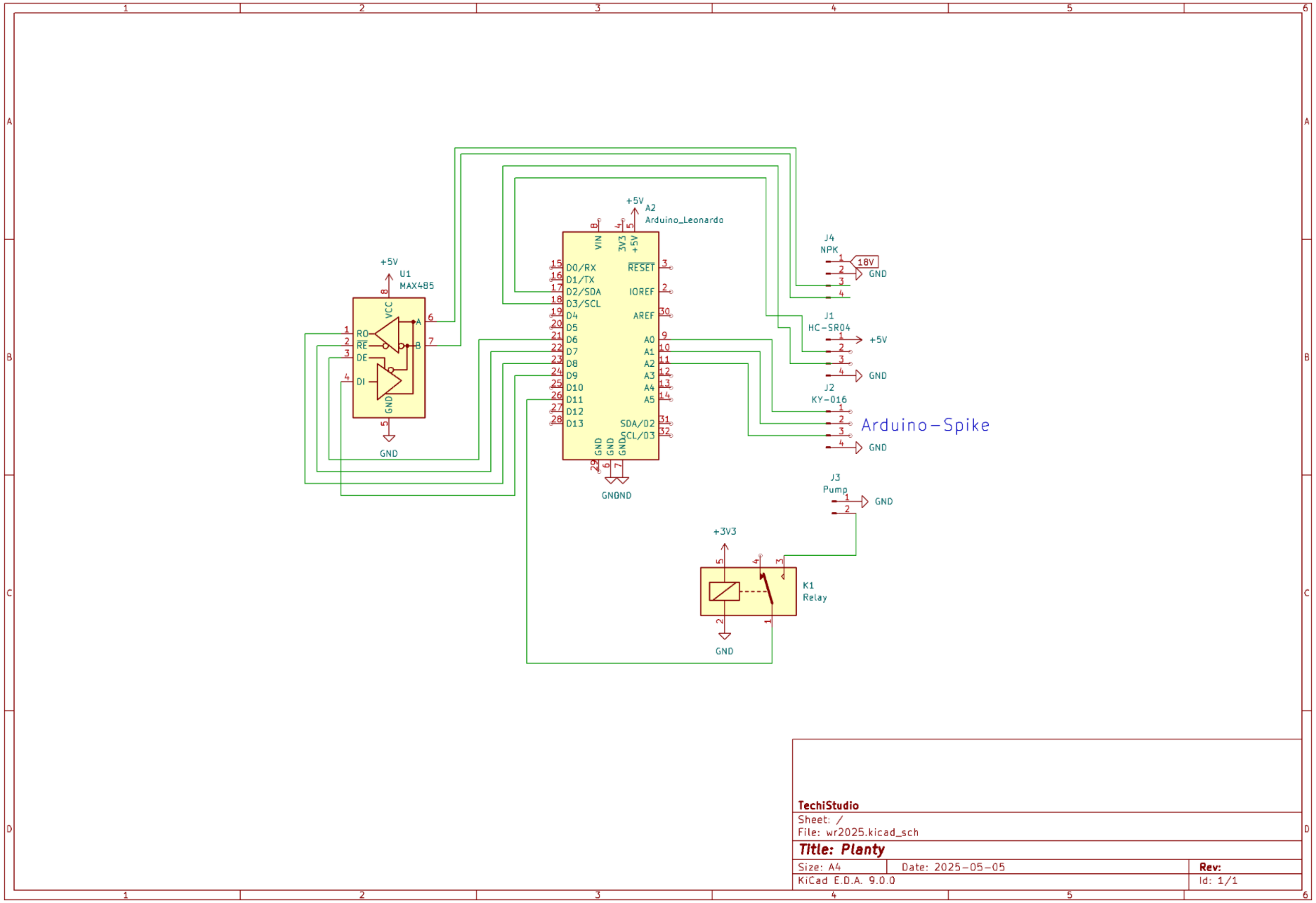
```
void npk_keyboard(int n, int p, int k) {
  Keyboard.write(KEY_F13);
  delay(10);
  Keyboard.print(n);
  delay(10);
  Keyboard.write(KEY_F13);
  delay(10);
  Keyboard.write(KEY_F14);
  delay(10);
  Keyboard.print(p);
  delay(10);
  Keyboard.write(KEY_F14);
  delay(10);
  Keyboard.write(KEY_F15);
  delay(10);
  Keyboard.print(k);
  delay(10);
  Keyboard.write(KEY_F15);
}
```

It also communicates what it had planted, with the keypresses of the F16, F17 and F18 keys. The recorded data can be exported as a json file.

The Spike

The program of the Spike is much less complicated, as it most of the time only moves the conveyor belt, so we'll only include our main program logic:





+5V
A2
Arduino_Leonardo
VIN
3V3
+5V
D0/RX
D1/TX
D2/SDA
D3/SCL
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
RESET
IOREF
AREF
A0
A1
A2
A3
A4
A5
SDA/D2
SCL/D3
GND
GNDGND
+5V
U1
MAX485
VCC
RO
RE
DE
DI
A
B
GND
J4
NPK
18V
GND
J1
HC-SR04
+5V
GND
J2
KY-016
Arduino-Spike
GND
J3
Pump
GND
+3V3
K1
Relay
GND
TechiStudio
Sheet: /
File: wr2025.kicad_sch
Title: Planty
Size: A4
Date: 2025-05-05
KiCad E.D.A. 9.0.0
Rev:
Id: 1/1

Innovation and research

Innovation

In the project, we took a lot of inspiration from vertical farms. According to some experts and of our own, this technology can help us make it easier to get food for future space explorers and astronauts. Our robot helps to solve the problem of relying on resources sent from earth while exploring an alien celestial body e.g. while exploring the Moon or Mars. Or if this opportunity cannot be used due to the composition of the regolith or other circumstances, it helps to map the composition of the celestial body over larger areas. These processes can be accelerated by the fact that the base can operate with more excavator robots. This technology can be valuable not only in space but also has a benefit on Earth. Because our system independently manages the planting and organization of the plants and prepares them to grow indoors in artificial conditions and therefore take up less space. It can be of high importance in urbanized areas, where it can save valuable free land and produce oxygen. It can be used in deserts that have not yet been used, where the heat can be used in greenhouses. For this reason, we believe that its implementation is a useful step towards achieving the end of hunger among the Sustainable Development Goals collected by the United Nations. Our solution would probably be popular among companies and small businesses, for whom it would simplify crop production, and only the products produced by it would be available to consumers. For more information about the business of our project refer to the business model canvas below. The project makes living long-term on foreign planets plausible and makes managing resources for the explorers easier, as the time spent maintaining a stable food supply is less and maintaining that food supply is less physically demanding. This can be also said for users on earth, as it makes agricultural work much less demanding, and because of this, farmers can concentrate on the well-being of their plants, not the demanding work of tilling, watering and planting them. It would also benefit consumers, as fruit and vegetables will be healthier and tastier, and because of the and easier work - cheaper.

Research

About the nutrients, conditions, and existing technologies required for the cultivation of plants we looked many different sources, mainly the USDA website and various scientific and agricultural articles, that you can find in the sources.

We also researched the geography and composition of planets, and their relationship with life. Regarding this, Vilmos Steinmann, the Martian researcher, geographer from the Astronomy and Earth Sciences Research Centre (CSFK), personally lectured us and shared scientific articles with us. For example, we learned the following important things: Water can only be found 5-10 km deep underground in the pores of rocks. With today's technologies it would be too expensive to get access to it. However, water can still land on planets constantly, for example by comets. Mars is on the border of the habitable zone. In the Martian equatorial area, the temperature is not as extreme as on other areas (Between +27 ° C -70 ° C), which will allow closed crop production in the future. We have received information on the composition and usability of Martian regolith (SOIL). It can support undemanding crops such as salads. These may be suitable for oxygen production, but not yet for human consumption due to the high arsenic and cyanide content of Martian rocks. However, it is possible to develop a technology to remove these toxic substances from the soil. We also learned a lot about how the Mars Exploration Robots work, their sensors and how they communicate with Earth. The composition of the rocks is analysed by emitting a laser beam on them. Samples are also taken from the regolith, which is later analysed. The results are transmitted back to Earth. For example, laser emission cannot be done frequently, because the reflected light heats up the receiver. The robot travels very slowly because it requires a back-and-forth exchange of messages with the Earth to control it, which takes between two times 3-22 minutes, depending on the current distance between the two planets.

The Business Model Canvas

Designed for: TechiStudio

Designed by: Andorfer Zalán

Date: 2025/04/15

Version: 1.0.0

Key Partnerships

- Manufacturing suppliers for parts manufacturing
- Possible joint development and support with and from Space Agencies
- Distributors for easier transportation

Key Activities

- Efficient and quality manufacturing
- Close customer relationships:
  - Extensive documentation
  - Well designed software

Key Resources

- Manufacturing equipment and personnel
- Research personnel
- Maintenance personnel
- Transportation equipment

Value Propositions

- Mainly quantitative:
  - cost reduction in agriculture
  - providing access to freshly grown food in space
  - easier usability than traditional farming equipment
  - better performance than traditional means

Customer Relationships

Dedicated personal assistance before and after sale maintenance and support.

Channels

Mainly selling using own channels and distributors, repairs and customer service through own channels

Customer Segments

- Niche market
  - Agricultural companies
  - Space Associations

Cost Structure

- Mainly value-based proposition
- Biggest expenses:
  - Research personnel
  - Manufacturing
- Using fixed costs
- Focused on economies of scope

Revenue Streams

- Asset (one-time) sales of the product
- Usage fees for:
  - repairs
  - maintenance
  - construction

Turn ideas into revenue with Strategyzer's innovation programs

Copyright Strategyzer AG
The makers of *Business Model Generation* and *Strategyzer*

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Strategyzer
strategyzer.com/innovation

Sources

- http://ars.usda.govhttps://www.ars.usda.gov/oc/utm/vertical-farming-no-longer-a-futuristic-concept/
- https://www.grandviewresearch.com/
- https://www.verticalfarmdaily.com/
- Wamelink et al, Open Agriculture, 2019
- Part of the NPK sensor communication code was interpreted from: https://forum.arduino.cc/t/soil-sensor-7-in-1-print-mixed-values/1067383/14
- NPK ratios:
    - Yin, Zc., Guo, Wy., Liang, J. et al. Effects of multiple N, P, and K fertilizer combinations on adzuki bean (Vigna angularis) yield in a semi-arid region of northeastern China. Sci Rep 9, 19408 (2019). https://doi.org/10.1038/s41598-019-55997-9
    - Malghani, abdul latif & Ullah, Asmat & Sattar, A & Hussain, Fiaz & Abbas, Ghulam & Hussain, Jamshad. (2010). RESPONSE OF GROWTH AND YIELD OF WHEAT TO NPK FERTILIZER. 24. 185-189.