Horizon 2020

flora **robotica**

Societies of Symbiotic Robot-Plant Bio-Hybrids as Social Architectural Artifacts

Deliverable D1.2

Evaluation of mechatronics prototype of the robotic symbiont including supporting software

Date of preparation: 2017/03/31	**Revision: 1 (r740)**
Start date of project: 2015/04/01	**Duration: 48 months**
Project coordinator: UPB	**Classification: public**
Partners: *lead:* **ITU**	*contribution:* **CYB, CITA, UPB, UNIGRAZ**
Project website:	http://florarobotica.eu/

H2020-FETPROACT-2014

DELIVERABLE SUMMARY SHEET

Grant agreement number:	640959
Project acronym:	*flora robotica*
Title:	Societies of Symbiotic Robot-Plant Bio-Hybrids as Social Architectural Artifacts

Deliverable N^o:	Deliverable D1.2
Due date:	M24
Delivery date:	2017/03/31

Name:	**Evaluation of mechatronics prototype of the robotic symbiont including supporting software**
Description:	Development of plant-robot hybrid organisms is a main driver in *flora robotica*. A broad range of hardware studies and developments has been investigated for such hybrids. Here, we present and evaluate the current hardware developments and their software control aspects. The consortium has limited the focus to braids as the basis structures for this robot symbiont, however, this still includes diverse investigations because these structures come with new challenges. These atypical robotic systems demand both novel approaches to material assemblage as well as novel approaches to control and actuation in order to develop and grow plant-robot hybrids in meaningful ways.

Partners owning:	**ITU**
Partners contributed:	**CYB, CITA, UPB, UNIGRAZ**
Made available to:	public

Contents

1 Introduction: Robotic Symbiont and the basis for evaluation

One of the main drivers of *flora robotica* is to examine the potential of a plant-robot hybrid organism and its potential implementations. From the Description of Work the project was narrowly focused on conventional robotic construction kits as the basis for the robotic symbiont. However, early in the project it became clear that the construction kits were lacking in terms of potential for interacting with plants. Hence, the consortium engaged in a broad range of hardware development in order to create a broad technical basis from which the robotic symbiont could grow in a more meaningful way. This phase ended around M18 and it was necessary for the consortium to focus on a few selected platforms to ensure we did not spread our resources too thin (a point also put forward by the reviewers in the evaluation report). The consortium decided to pursue all technologies that support the vision of braids as the basis for the robotic symbiont while all other activities are either stopped already or are about to be finished (i.e., work that was started before M18 is going to be finished but no new sub-projects are going to be started) as indicated in Table 1.

Prototype	**Role**	**Partners**	**Status**
Braiding robots	robots producing braids	ITU	active
Braided robots	actuated, intelligent braids	CITA, UPB	active
PBDW nodes (decision wall)	guidance for growth in braids	UNIGRAZ, CITA, UPB	active
Modular robots	energy management in braids	ITU	active
MU sensor	plant sensing for robotic nodes	CYB	active
Swarm bots braiding (Thymio II)	mobile robots producing braids	CITA, UNIGRAZ, UPB	finishing
Assembly of robot modules (Thymio II)	actuated, intelligent braids	UNIGRAZ	finishing
Swarm bots formations (Kilobot)	embodied swarming simulation	UPB	finishing
LocoKit	mechatronic structural basis	ITU	stopped

Table 1: An overview of the active, about to finish, and stopped hardware efforts in *flora robotica* as a result of narrowing our focus.

1.1 Structure of the reported work

We are actively working on four aspects of creating an eco-system of braided robotic symbiont technologies:

- (Sec. 2.1) braided robots and intelligent filament,
- (Sec. 2.2) sensor-actuator nodes to interface a braided symbiont to the plant symbionts,
- (Sec. 2.3) technologies related to the automatic production of braided structures, and
- (Sec 2.4) technologies exploring the potential of energy autonomy of a robotic symbiont.

While the first two are technologically compatible the third is a prerequisite for the vision to be feasible on a larger scale and the fourth is currently investigated with independent hardware, but with an aim to integrate the results of this work into the eco-system of braid technologies.

	Braided robots	Braiding robots	PBDW nodes	Modular system
scalable	✓	✓	✓	
load-bearing for plants and structures	✓			
soft and extendable	✓			
passive plant guidance	✓			
provides actuation	✓			
distributed control	✓		✓	
produceability		✓		
active plant guidance			✓	
plant sensing			✓	
energy autonomy				✓

Table 2: Overview of what elements the work on the various prototypes provides to the overall vision of braid-based robotic symbionts.

Table 2 gives an evaluation of how the four aspects are sufficient and necessary to support the vision of a fully integrated braid-based robotic symbiont. This deliverable presents the current status of these four strands of hardware work and their supporting software.

1.2 Braids as mixed structural, kinetic and computational substrate

Explorations with braided robotic symbiont technologies have demonstrated a number of potentials. First, braided structures can form a number of different topologies with various attributes amongst other cylindrical structures and structures with bifurcations, holes, and sockets (see Fig. 1). Second, it was demonstrated that structures could be braided to attain specific kinetic behaviors through specific interweaving patterns. Third, braids can be composed of mixed materials resulting in gradation of attributes between areas of different material within braids.

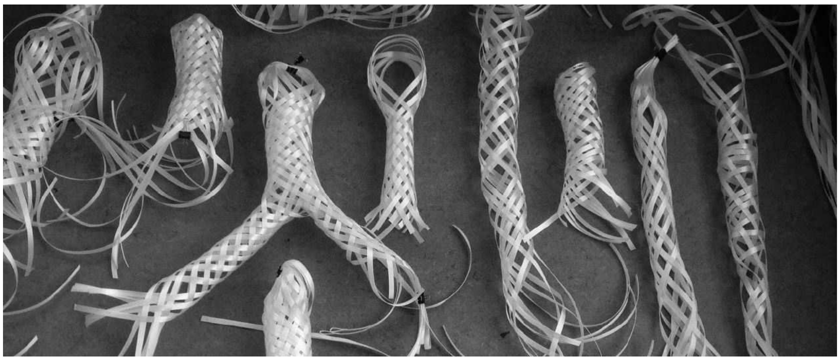


Figure 1: A variety of braid structures can be constructed by changing only the interlacing pattern.

Fourth, single strips of a braid can be understood as independent sensor and actuated elements, each with robotic potentials. These elements can act at the lowest component level of the braid. As such, a braid comprising of individual active elements can act as a de-centralised system.

These aspects combined increase the potential for interaction between braided structures and plants. Based on this conclusion the consortium decided to focus the braid logic as structural, kinetic and computational base substrate. The consortium now works towards this vision through the above mentioned technologies, which are investigated through a number of experimental mechatronic prototypes.

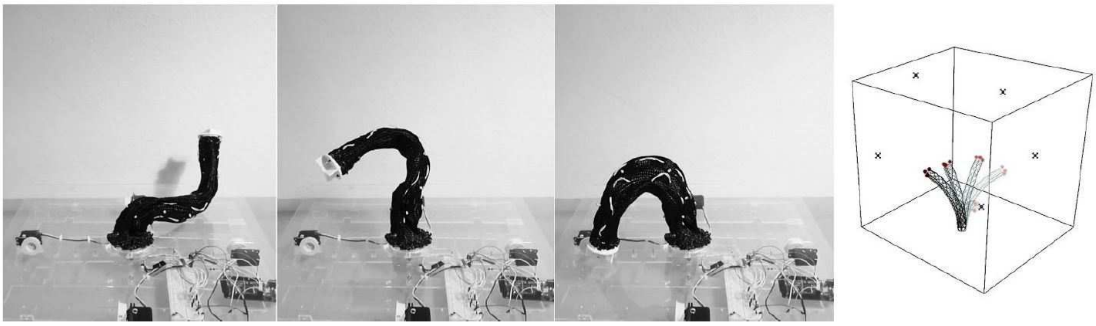


Figure 2: First iteration of an actuated braided robot; left: braid arm with triaxial actuation strings attached to pulley motors, in three distinct deformation states; right: model of the triaxial actuation string setup, for use in evolution in simulation.

2 Description and evaluation of mechatronics prototypes

2.1 Actuated braided robots, supporting software, and intelligent filament

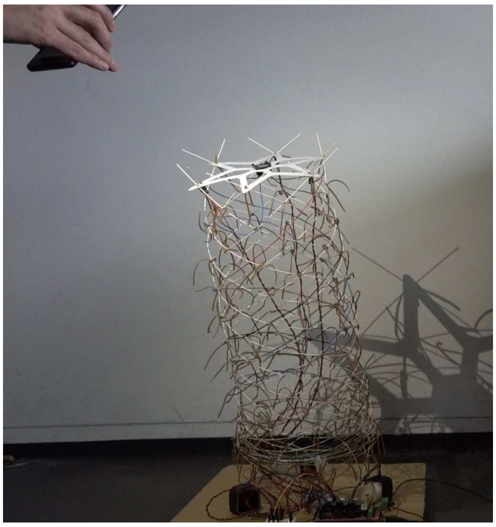
2.1.1. First generation actuated braid robots

The first prototype of an actuated braid (see Fig. 2, left) follows the scale and actuation approach of the Pot-arm. Contrary to the Pot-Arm – that was based on a tensegrity structure using elastic bands as tension and pot-shaped shells as compression – this prototype is made from industrially braided nylon filament at the length of about 400mm and 30-60mm of diameter. The nylon braid is attached to a base plate and reinforced by four glass-fiber rods going directly from bottom to top inter-weaving the nylon braid. For actuation this prototype has three strings also interwoven triaxially that connect the top piece of the structure to three individual servo motors. As the actuation strings are individually tightened or loosened the nylon braid and rod assembly is elastically deformed by the shifting forces (see Fig. 2, left). To achieve specific elastic deformation in the structure the strings can thus be tightened with various strengths making the tip of the prototype point in different directions. However, because the structure is made from a mix of materials and elements the deformation is erratic and ill-defined.

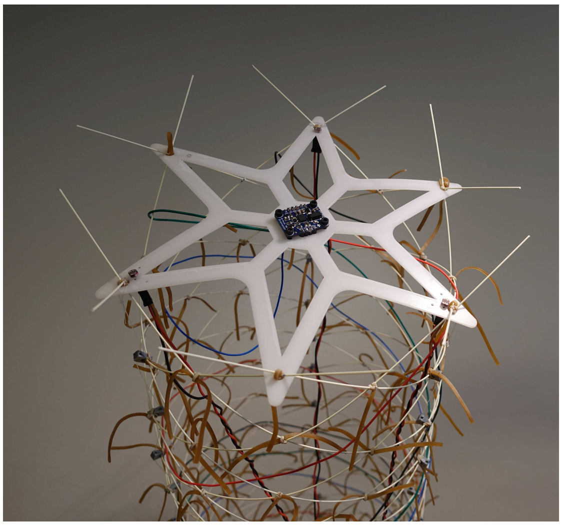
2.1.2. Second generation actuated braid robots

The second generation actuated robot prototype is braided from 1mm Glass Fiber Reinforced Polymer (GFRP) rods. This soft braided robot tower, or Braidbot, is 1 meter tall and 20 cm in radius (Fig. 3(a)). To maintain its shape during deformation the rods are fixed at the node intersections. It makes use of the same principles of actuation as the first generation soft braided robots. However, this new generation is actuated by three NEMA 13 stepper motors positioned at 0, 120 and 240 degrees off the base radius of the structure. Each motor extend and retract a nylon string that is weaved triaxially through the GFRP braided structure and ties to the structure's end element, that we call 'crown' (Fig. 3(b)). The extension and retraction cause the structure to bend in the plane parallel to the motor. The crown carries three LDR sensors aligned with the steppers motors along with a 9-degrees-of-freedom IMU sensor. The IMU is capable of providing absolute orientation of the tip of the structure.

Low level control of the robot is achieved through an Arduino MEGA 2560 with a RAMP



(a)



(b)

Figure 3: (a) Braided robot tower (Braidbot) performing follow light behavior, (b) Braided robot tower crown with three LDR sensors aligned with motors and 9DOF IMU sensor in the center.

1.4 shield connected to both sensors and stepper motor actuators. The behavior of the Braidbot can be programmed to be prescriptive by the user setting multiple positions of the robot pre-established, or reactive, where the robot uses it sensing capabilities to follow or avoid a light source detected by LDR sensors in the crown.

The Braidbot draws upon the advantage of braided structures and contributes to the field of soft robotics by its lightweight combined with embedded material performance. This second generation also contributes to the diversity of braided structures in the project. Future investigations are being examined in collaboration among various partners such as the integration of a neural-network-based learning mechanism that would allow for a non-linear control of the robot (machine learning) and the integration of plant-sensing capabilities to the robot that could potentially lead its control by the plant itself.

2.1.3. Intelligent filament: collective decision-making on braids

In *flora robotica* we want to expand the braided structures with sensors and computational devices to create 'intelligent strips'. Fig. 4 shows a preliminary embedded-sensor filament. In future work, we plan to have many active filaments (with sensing, processing, and actuation) able to sense, collectively decide, and act.

In this electronics-embedded soft-body robot context, we aim at achieving distributed decision-making by including braid filaments with embedded sensors. On our way to fabricate and assemble parts of the intelligent strips into real life prototypes, we work at the same time on the distributed algorithms that will run on the strips in the future. Multiple crossing strips, each equipped with sensors, communication, and computation units, form a swarm of static units. There will be streams of data that are either acquired by the sensors or communicated via the other units. The process of deciding on a behavior accordingly will demand a collective decision-making procedure similar to the one in a swarm but this time a stationary swarm. One of the collective decision-making methods is Local Majority Rule (LMR) that can be a viable solution

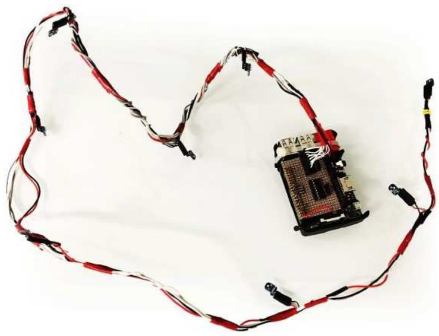
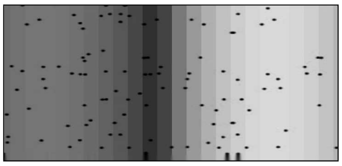
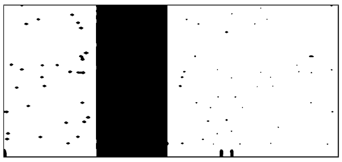


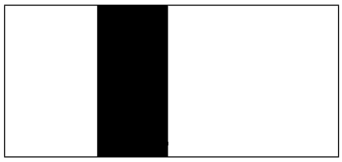
Figure 4: A simple prototype of intelligent strip with a Raspberry Pi and eight ambient light sensors along the filament.



(a) a flattened cylindrical braid



(b) filtered bitmap



(c) final output

Figure 5: (a) an initial bitmap of the light perceived by the sensors is shown, (b) the filtered bitmap after applying the threshold, (c) the result of LMR.

when every unit has a limited and local information. In a simple LMR, the units will count the votes from their neighbors and turn their states into the one from the majority.

We solved a simple problem of reaching consistency on the light presence among a set of fault-prone sensors. Two geometrical braided compounds (i.e., planar and cylindrical shapes) were studied. Shade formation procedure is different in cylindrical and planar braids. We assume that in the planar braids, all sensors receive the same amount of light regardless of their position on the plane, if there is no shield to block the light. In cylindrical braids, however, one side receives more light (bright) than the other (dark). The light projection on the cylindrical braid varies from time to time (simulating the presence of sun as the light source). Nevertheless, the algorithm operates regardless of the procedure of light projection.

If the braid is composed of only intelligent strips (no passive filaments), a bitmap can model the perceived light by the ambient light sensors. Each pixel of this bitmap is a value in $[0, 255]$, where zero represents darkness and 255 means that the sensors receive the maximum intensity of light. Fig. 5(a) is an example of such a bitmap. We then apply a threshold to filter the range $[0, \theta)$ as dark and $[\theta, 255]$ as bright (see Fig. 5(b)). Noise in our scenario is the failure in the process of capturing the light. Note the black spots in the bitmap due to these noises. Running the LMR algorithm on the bitmap with a local neighborhood of size one divides the bitmap into two distinct sides. We do not desire a 100% consensus since the division in the swarm represent the dark and bright sides of the cylinder. Nevertheless, the current method only filters out small blocks of noises. In future work, we plan to have learning mechanisms to improve the decision-making process.

2.2 PBDW sensor-actuator nodes: distributed robots to interface natural plants

The second aspect mentioned in the introduction is the approach of sensor-actuator nodes to interface a braided symbiont to the plant symbionts. We have developed robotic nodes that sense the presence of a plant and interact with it via light.

In a second, parallel effort we develop an electrophysiological system, that includes tissue impedance and bio-potential measurements. We plant to integrate these approaches in one robotic node later in the project.

The description of robotic nodes in this section refers to our submitted paper on distributed robots for plant control [12]. We explore alternative methods of additive manufacturing by using distributed robots to steer plant growth. We investigate our robotic nodes in the Plant Binary Decision Wall (PBDW) setup. Although plant growth is slow, the addition of material can be almost for free and in certain scenarios the speed of growth is not problematic. The plant's natural adaptive behavior can be exploited by expressing appropriate stimuli. From a potentially long list of possible options (e.g., mechanical, chemical, electromagnetic), we select certain spectra of light as stimuli.

Despite our ambitious vision, we have to start simply to explore early concepts. With intention, we chose a simplified scenario that is technically feasible with technology of today, but also still contains the major challenges we will face in later work towards our vision. We grow plants at a plane; that is, we reduce the problem to 2-d instead of 3-d to limit the possible directions of growth. In addition, we place our robots in a predefined pattern − a diagrid − to further reduce the complexity (see Fig. 8(c)). In the experiments reported here, we keep the robots static. Currently, we explore methods to implement slow speed robots (e.g., 'slowBots' [1]) that could move in sync with the plants' growth. We use climbing plants and grow them along the diagrid to limit the plant's decision to a binary left-vs-right decision that is easily observed and easily verified. In our chosen initial setup, we can showcase the essentials of our methodology and address the major challenges of our project. This lays the groundwork for future extensions to braided structures, as detailed in Sec. 2.2.7.

2.2.1. Challenges of shaping plants with robotic stimuli

As described in [12], for the required implementation of distributed robots that steer plant growth, we have identified robotic challenges in four sub-domains of this work: sensing, actuation, integration of robots into the physical structure, and running robot experiments and plant experiments combined in one implementation. The sensing challenge is to detect the proximity of the plant tip, as our robots have to autonomously react to the current state of the plant. We have extensively tested different approaches and converged on using infrared sensors to detect plants that have grown into the sensor's coverage. Although the climbing plants grow along rods, and hence the direction that needs to be monitored is determined, the sensing challenge remains (e.g., small size of plant tip, dynamic light conditions). The experiment is visually monitored, but imaging is not used as a sensor because we want to follow a decentralized approach, without (sensing-)modules external to the robot. This is due to the agenda and requirements of the application *flora robotica*, that aims for easy-to-use products that can be replaced by non-expert users and operate in unknown environments.

The actuation challenge is to send stimuli that trigger appropriate plant behaviors. Plants are sensible to certain changes in their environment. Stressed plants would slow down or even stop their growth, and eventually would die. While plants suffer in conditions with too limited light, there can also be too much light. Hence, we follow the biological literature [5, 4, 7], and we have extensively tested our setup with different light conditions. We use red light to support the plant's healthy photosynthesis and blue light to trigger a behavioral response in the plant that, as an effect, changes the direction of growth.

The challenge of integrating the robots into the diagrid structure may seem trivial but there are several requirements that made preliminary experiments necessary. The overall structure needs to be climbable by the plant, meaning plants need to be able to bypass the robotic nodes that must also serve a structural role. The distances between junctions in the diagrid should be short to limit the time required for growth experiments, but cannot be too close to each other because then plants might randomly grow into several different rods.

The challenge of combined robot/plant experiments is to merge standard experiment protocols from two different fields of science. The usual robot lab is not designed to grow plants and the usual plant lab is not designed to do robot experiments (similarly for the experimenters). We had to select the right plant species that grow fast enough to make the experiments feasible but also grow reliably in a robot lab. Appropriate environmental conditions (e.g., ambient light, temperature, humidity) have to be provided and then guaranteed constantly for a long period of time (eight weeks and more). Plant growth is slow and therefore the experiments have long durations. In mobile robotics, one is used to certain rapid-prototyping methods in experiments that even allow for a fast trial-and-error procedure when necessary. This is not available here as the experiments are costly in overhead, predominantly in terms of time.

2.2.2. Robotic nodes for plant shaping

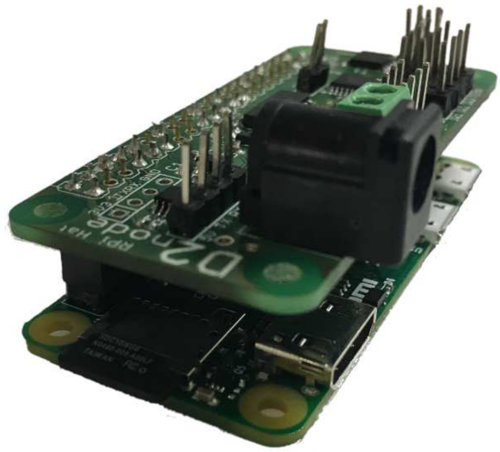
As described in [12], we investigate how to shape natural plants by an autonomous, distributed robot system (more information about the plant symbiont in deliverable *D2.2 Report on the final algorithms and plant-affection of bio-hybrid organism*). Here, we present initial steps towards this goal by introducing our distributed system of robotic nodes. An individual robotic node is approximately half-spherical in shape (≈ 9 cm $\times 10$ cm in width and height, with a flat back), contained in a polymer 3-d printed case that is faceted according to the orientation vectors of various sensors and actuators (see Fig. 6(a)). It consists of photoresistors[1], GP2Y0A41SK0F IR-proximity sensors[2] for plant tip proximity detection (see D2.2 for more information), far-red[3] and RGB LEDs as light stimuli (see D2.2 for more information), a Wireless Local Area Network (WLAN) module[4], and a Raspberry Pi Zero[5] with our custom Raspberry Pi HAT[6] (see the Appendix and Fig. 6(b), PCB interfacing the included sensors and actuators to the Raspberry Pi header).

In the current setup, we use WLAN for node-to-node communication, but we emulate local communication by allowing the robotic nodes to only communicate with direct neighbors. The information communicated is detection of plants. In future experiment setups, the photoresistors will instead be used for this task to implement a decentralized communication system.

The software implementation regulates uploading sensor data and log files to a Networked Storage Device (NAS). The ZeroMQ[7] library is used to implement communication among on-board processes and with direct neighbors. ZeroMQ is also used for logging via the network, allowing a user to visualize the stream of sensor data at run time.

[1] 5mm photoresistors datasheet http://ronja.twibright.com/datasheets/cds-resistor-pgm.pdf
[2] Sharp GP2Y0A41SK0F IR proximity sensor datasheet https://www.pololu.com/file/0J713/GP2Y0A41SK0F.pdf
[3] Far Red LED datasheet: https://cdn.shopify.com/s/files/1/0920/1206/files/Future_Eden_Far_Red_LED_740-745nm_Data_Sheet_60ec003d-6e8a-438a-bc3e-4789acd5ab1e.pdf
[4] EDIMAX N150 WIFI module datasheet https://cdn.sparkfun.com/datasheets/Wireless/WiFi/EW-7811Un_Datasheet_English.pdf
[5] Raspberry Pi Zero: https://www.raspberrypi.org/products/pi-zero/
[6] Raspberry Pi HAT: http://www.cybertronica.de.com/download/D2_node_module_v01_appNote16.pdf
[7] ZeroMQ asynchronous messaging library http://zeromq.org/

(a) Current robotic node design



(b) Raspberry Pi HAT

Figure 6: Robotic node, Raspberry Pi HAT mounted on Raspberry Pi Zero.

The geometry of the robot body is designed to allow plants to surpass it, though it serves as the structural node of diagrid intersections. The domed, faceted shape allows the plant tip to incrementally find its way across the surface, allowing it to more easily reach the blocked stimuli. The individual faces of the body are oriented according to the tasks of the various sensors and actuators. Each node has four attachment points for scaffold rods that are coplanar (in the four corners of the node) and one attachment point for a 3-d scaffold rod projecting out of the diagrid plane. For current Binary Decision Wall experiments, only coplanar diagrid scaffold rods are included (shown in the setup of Fig. 8(c)); attachment points for '3D' rods are included for potential use in later work. The scaffold rods attach to the robotic nodes at a 45° angle. Near the attachment points there is a photoresistor. Near the attachment points of each rod where plants may approach (the two lower diagrid rods, and the one 3-d rod) is also an IR-proximity sensor and a far-red LED. Each photoresistor, IR-proximity sensor, and far-red LED is embedded in a planar face with its z-axis aligned to the vector of the respective scaffold rod (see Fig. 8(a)). The six RGB Pixies are arranged in a triangular layout, oriented around a dome-like faceted surface. They are positioned and oriented such that they provide an equal distribution of light to the three directions of possible approaching plants (see Fig. 8(a)). In this way, the six total LEDs provide each of the three lit directions with three light sources, such that six LEDs can do the task of nine. This allows the light source for each direction to be spread over three LEDs rather than two, distributing the power consumption and problematic heat load. The geometry of the 3d-printed case, in combination with the location and orientation of the photoresistors, contributes to blocking the photoresistors from sensing light cast by their own node's RGB LEDs.

2.2.3. Experiment setup with Plant Binary Decision Wall

As described in [12], for our experiments using the robotic nodes in a bio-hybrid system, we arrange the planar mechanical scaffold in a diagrid layout, in the Plant Binary Decision Wall (PBDW). The structure is arranged such that there are four potential intersections where a plant must make a binary decision (left or right) about its next growth direction.

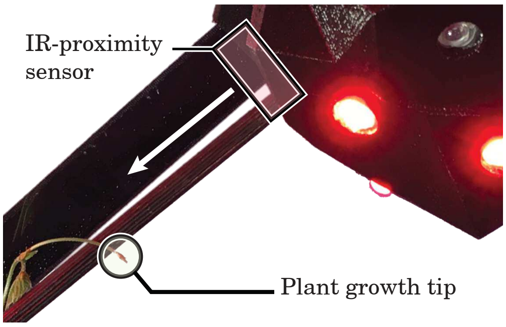


Figure 7: Plant tip detection.

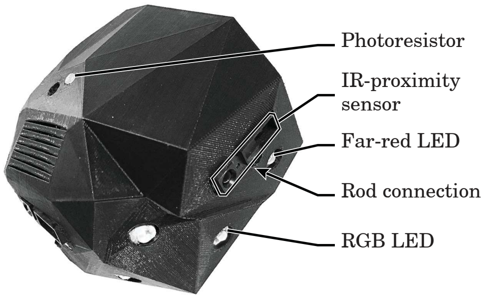
The PBDW contains eight robotic nodes, distributed in four columns and four rows, on a geometrically regular diagrid scaffold. For current PBDW experiments, only 2-d coplanar rods are used, though the robotic nodes also include hardware and connection points for 3-d rods (for potential use in future work). Our PBDW setup is 125 cm × 0.5 cm × 180 cm in width, depth, and height. The robotic nodes are connected with 40 cm rods, oriented according to the geometric constraints in Sec. 2.2.2.

In addition to the scaffold carrying the robotic nodes, there are two Global Environment Monitors (GEMs). The primary GEM is operated by a Raspberry Pi 3 and equipped with a BME280 temperature-pressure-humidity sensor, a TCS34725 RGB color sensor, a soil moisture sensor, a water pump, and a camera module. Hence, it is capable of monitoring the room environment where the PBDW is placed, provide the plants with water when required, and capture close-up time-lapse videos of the experiments. In Fig. 8(b), the location of the sensors and water nozzles is shown. The secondary GEM is a simpler version; it captures time-lapse videos from a different angle, capturing the full wall. Similar to the robotic nodes, the GEMs upload the sensor data and log files to the NAS. Both GEMs are running a reply server that can answer specific requests (e.g., the latest measurements of a specific sensor).

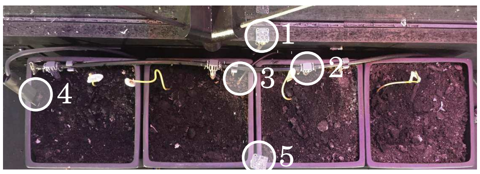
Two 45W ‘Erligpowht’ LED plant growth lamps[8] are added to the system to provide sufficient amounts of constant monochromatic red light, assuring the survival and health of the plants during the experiments. An ‘Erligpowht’ growth lamp contains 225 LEDs, 165 red and 60 blue, with peak-emissions λ_{max} at wavelengths 650nm and 465nm respectively. The blue LEDs on both lamps were concealed to prevent any interference with the phototropic blue light stimuli imposed by the robotic nodes. At each side of the PBDW, we place one growth lamp at 90 cm height and 60° inclination facing the plants (as shown in Fig. 8(c)).

We define two states for the robotic nodes: *guiding*, the node emits only blue light to attract the plants and detects the proximity of approaching plant tips (proximity sensors are on); and *feeding*, the node emits only red light to support the plants’ photosynthesis (proximity sensors are off, no plant tip detection). First, we run control experiments where all robotic nodes are set to *feeding* state to show the natural behavior and growth of plants without imposing light stimuli. Second, we show the ability of the robotic nodes to guide the plant growth into a predefined pattern exploiting the plants’ behavioral responses to blue light (via the *guiding* state).

[8]‘Erligpowht’ LED plant growth lamps
http://led-growlights.net/product/top-erligpowht-45w-led-red-blue-indoor-garden-plant-grow-light-hanging-light-price



(a) Components of the robotic node



(b) Pots with plants after 26 hours, top view.

(c) Plant Binary Decision Wall (PBDW)

Figure 8: The robotic node, GEM sensors and actuators, PBDW experiment setup. In (b), labeled components of GEM: 1. air temperature, pressure and humidity sensor; 2. water nozzle; 3. soil temperature sensor; 4. soil moisture sensor; 5. RGB color sensor.

2.2.4. Results of control experiments

As described in [12], we have conducted two control experiments over a total time period of about three weeks. Both experiments included four plants. The climber plants showed unbiased upwards growth due to the lack of a directional phototropic stimulus by blue light, in addition to standard gravitation that drives plants upwards (see Fig. 9(a)). The typical winding motion behavior was observed, a plant behavior to explore the environment and search for climbing support. However, the plants were not able to attach to the location and orientation of our scaffold rods without further external steering. The experiments were stopped when at least two plants grew to a height that caused them to collapse without support due to their own weight[9] (see Fig. 9(b) and Fig. 9(c)).

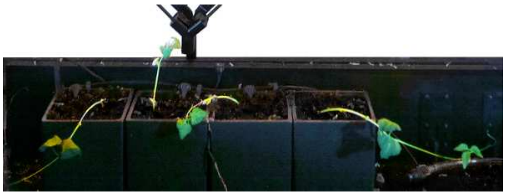
2.2.5. Results of predefined-pattern experiment

As described in [12], we investigate the ability of our distributed and decentralized robot system to grow plants in predefined shapes. Every robotic node in the system knows its location in the diagrid map and the required final pattern (z-shaped pattern, see Fig. 10(a)). Only one node is allowed to be in the *guiding* state at any given time. When the experiment starts, one node in the first level sets its state to *guiding* according to the predefined pattern. Once the node detects an approaching plant, it notifies its direct neighbors and switches its state to *feeding*. According to the pattern, the relevant node in the next level will switch its state to *guiding*. This process is repeated until the node in the final level detects an approaching plant.

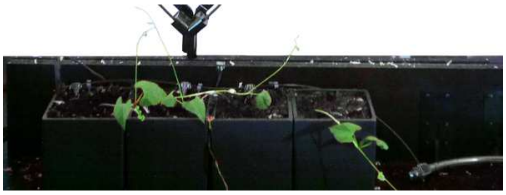
We have conducted two predefined-pattern experiments in total over a time period of approximately 14 weeks. The plant behavior was similar in both experiments. However, we explain

[9] video online: https://vimeo.com/205727160

(a) Control experiment #2, straight growth, after 5 days.



(b) Control experiment #1, 3 plants collapsed, after 15 days.



(c) Control experiment #2, 2 plants collapsed, after 7 days.

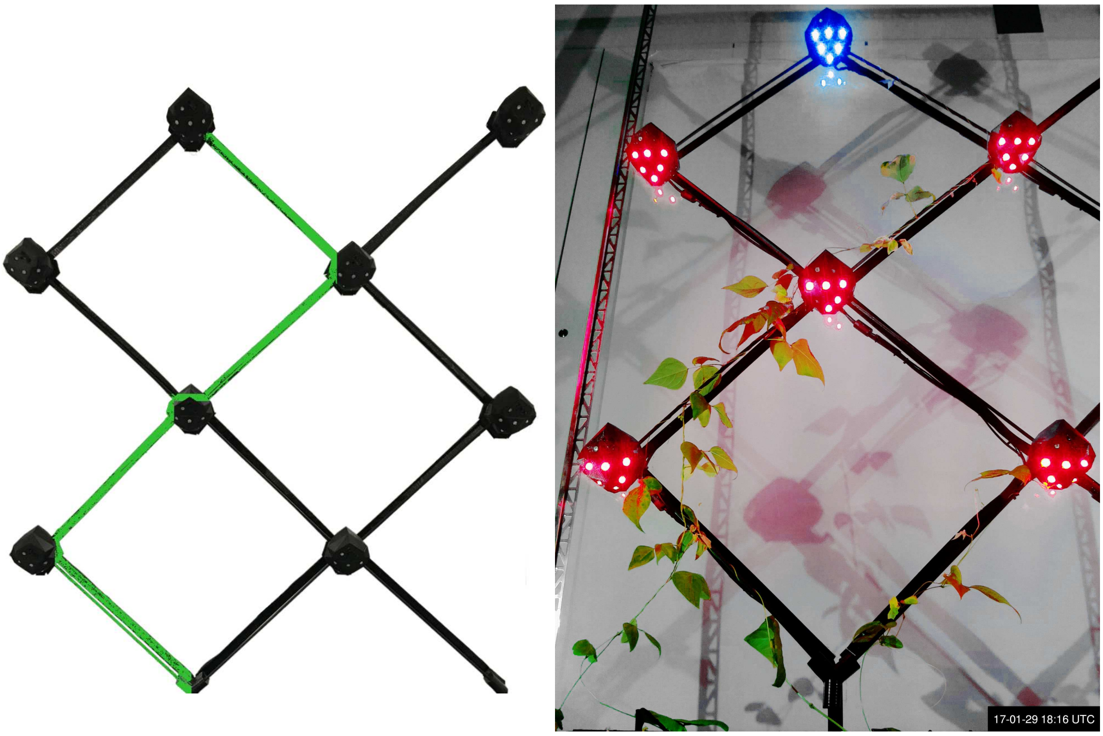
Figure 9: Frames of results from control experiments.

the plant behavior in predefined-pattern experiment #2[9] in detail. First, one plant succeeds to attach to the scaffold rod and climb along it. The robotic node detects the plant's presence when it is within range and triggers the node on the next level. Two more plants attach to the first rod then continue climbing towards the *guiding* node. However, the plant located at the far right fails to find the rod and continues to grow upwards until it collapses due to the lack of support. The three other plants continue to climb the scaffold and trigger a *guiding* node on the third level. Now, the plants have two choices: continue climbing along the first scaffold rod leading away from the blue light or climb around the robotic node to reach the scaffold rod that leads towards the blue light. Surprisingly, the plants grew around the robotic node, reaching the *guiding* node on the third level. Meanwhile, the collapsed plant grows a new tip, which attaches to a first level scaffold rod and grows towards the blue light. Finally, one plant tip arrives at the final destination on the fourth level, hence, the desired pattern has been grown (see Fig. 10(b)), and we stop the experiment.

In summary, we find a significant difference between the results of the experiments with a predefined-pattern and the control experiments. Hence, our method is effective in shaping a plant.

2.2.6. Expected future extensions for hardware and control

As described in [12], we will extend our approach by incorporating other stimuli in the control process in addition to visible light. We will investigate far-red LEDs as repellents for plants, as combining attractive blue light and repelling far-red light could allow more sophisticated control of shaping. In addition, we investigate other types of sensors (e.g., sensing sap flow, activity of the photosynthesis) that will possibly be added to the robotic nodes. We also test modifications to the design of the robotic nodes in order to build smaller and light-weight versions. They would then be more suitable to be attached to different types of structures including flexible scaffolds and bigger plants. In the current work, the plants do not serve a structural role, as they are fully supported by the mechanical scaffold. We will investigate climbing species that become woody over time, which could be shaped during their green phase and stiffened in their woody



(a) Z-shaped pattern

(b) Final configuration

Figure 10: Main result, (a) desired user-defined Z-shaped pattern, (b) photo of final configuration, predefined-pattern experiment after 40 days.

phase. Such species allow a changing relationship between scaffold and plant, where the scaffold serves as supporting structure in early growth stages and the plants serve as structure in later stages, creating bio-hybrids that can construct architectural artifacts. A definite next step of this research is to switch from 2-d to 3-d environments which is already intended in the current robot design. More complex shapes and desired patterns of green patches will be generated by applying several stimuli and developing extended controllers of the nodes. Extensions to the control will allow the robotic nodes to adapt to measured environmental conditions. For example, the intensity of the light emitted by every node can be adapted to the measured ambient light. The objective is to shape the plant regardless of variations in the light conditions. Fully decentralized algorithms will run on the robotic nodes allowing the system to act as a swarm that adapts to manual rearrangements of robots. The user can change the topology of the robotic nodes by adding and removing them. The photo-resistors will be used for local 1-bit communication to provide minimal spatial information. A decentralized algorithm producing appropriate patterns of pulse signals will then combine the information leading to an adaptable self-awareness of the diagrid topology.

2.2.7. Discussion of extension to braided structures

An important impending extension of the robotic nodes and the PBDW setup is to integrate with braided structures instead of a strut and node style scaffold. A planar diagrid structure,

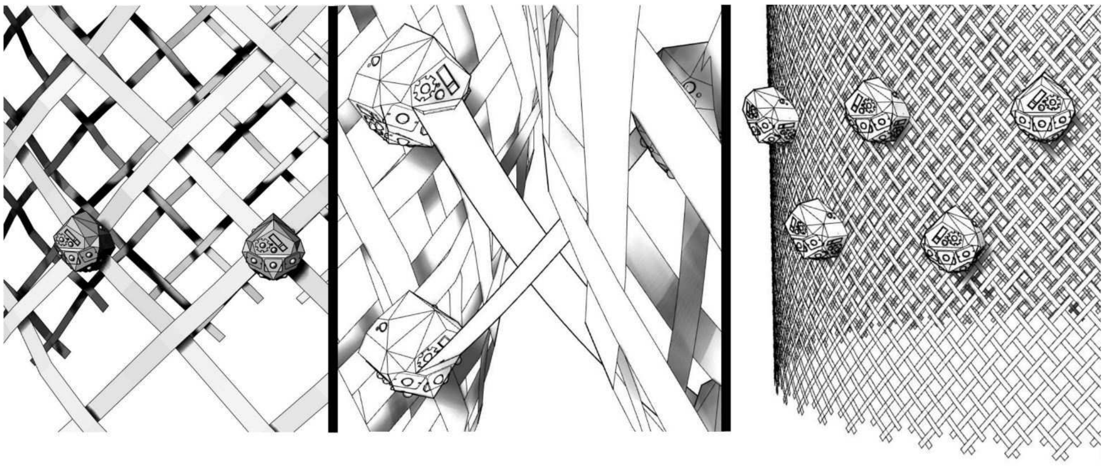


Figure 11: Sketches of integrating robotic nodes of the PBDW with braided mechanical scaffolds. Left, a Plant Binary Decision Braid; center, an extension to three-dimensional decision-making; right, an extension to gradients of decision-making.

like the one used in our current PBDW experiment setup, is easily analogous to a planar braid sheet. Therefore, a straightforward way to extend the diagrid scaffold to a braided structure is to replace the diagrid rods with braid filaments that have similar spacing and overall dimensions, locating the robotic nodes at the intersections of the filaments (see Fig. 11, left). Such a setup requires that the distance of filament between two intersections be substantial, resulting in a large sized braid. Because the circumference of a cylindrical braid would be quite large in this setup, its structure would be similar enough to the planar diagrid PBDW that the plants can be anticipated to grow along it in a similar way to the diagrid PBDW. The setup would be a Plant Binary Decision Braid, rather than a Plant Binary Decision Wall.

From this starting point, the setup can be extended from 2D binary decisions to 3D decision-making by utilizing the 3D connection point already included in each robotic node (see Fig. 11, center), as well as incorporating more connection points in future node designs.

Another crucial extension with braid is the move from discrete decisions to continuous gradients of decisions. This can be achieved in a braided structure by placing the robotic nodes at a few intermittent filament intersections on a very densely woven braid (see. Fig. 11, right).

2.2.8. Plant electrophysiology for sensing and measuring units

In addition to the development of the robotic nodes for the PBDW experiment with their IR sensors reported above, we also develop additional, more sophisticated sensors. This section is dedicated to results related to sensing, in particular electrophysiology.

It is challenging to interface with plants. Besides mere environmental sensors, we need to measure the actual plant responses in order to understand their internal processes and state. The options of getting feedback from the plant are limited but one of these options is electrophysiology. We distinguish two methods of electrophysiological measurements: bio-potential measurements and impedance tissue measurements. Using electrophysiology we get closer to the idea of controlling plants and to get live updates about their physiology. In addition, we also develop leaf transpiration sensors and sap flow sensors (not reported here).

The development of software and hardware for electrophysiology has been already reported in earlier deliverables. Here, we focus on a report on results from experiments and a discussion of experimental physiological data (based on different stimuli). See the appendix for an application note of the developed D2 node.

The electrophysiological system includes tissue impedance and bio-potential measurements. For measuring the impedance, the measuring system uses an auto-balancing bridge, where a test system is excited by the voltage V_V, see Fig. 12. The signal waveform for V_V is generated

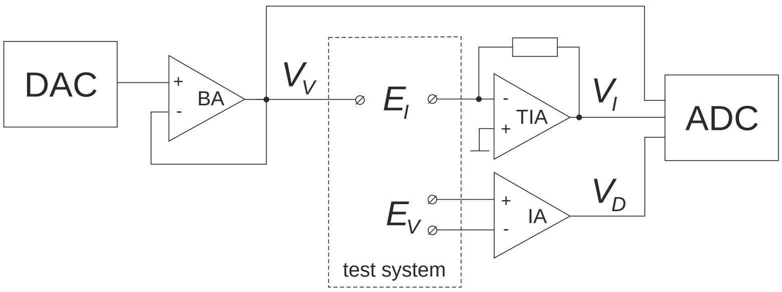


Figure 12: The measurement scheme, see descriptions in text.

by DAC and is buffered by the amplifier. The current I is converted into a voltage V_I by the transimpedance amplifier. Synthesis of the signal V_V occurs by Direct Digital Synthesis with 32-bit frequency resolution, the signals are digitalized by two synchronous 1.2 msps ADCs for simultaneous sampling of V_V and V_I signals. The measuring system uses an external analog circuitry for impedance matching. The electrode pair E_I is utilized for the current sensing $V_V \rightarrow V_I$ (so-called two electrode system). Another electrode pair E_V is used to sense a differential potential with the instrumental amplifier, this represents a so-called four electrode system. The MU allows using harmonic and non-harmonic signals V_V for driving an electrochemical system (e.g., for fast EIS). The impedance measurements do not use the window functions, this allows avoiding specific errors of this approach. The potential input with E_V electrodes has very high input impedance (input bias current is about $\pm$70pA), this enables sensing of bio-potentials in electrophysiological measurements. In such applications, the current can be also used for electro-stimulation purposes.

The measurement and electro-stimulation electronics are combined at the embedded measurement unit[10] and were developed by Cybertronica partner. The plant with electrophysiology measurement system is shown in Fig. 13.

[10] Measurement Unit EIS Impedance spectrometer http://cybertronica.de.com/?q=products/MU-EIS-spectrometer

Figure 13: The plant electrophysiology measurement setup.

2.2.9. Electrophysiological experiments with different stimuli

The plant reacts in a different manner at different types of environmental stimuli. The idea is to gather the experimental electrophysiological data – the reactions of different plants for different stimuli to determine and classify these impacts. In our initial electrophysiology measurements cacti and Dracena plants were used. The following figures show the Dracena plant's reaction for different stimuli, such as light impact (Fig. 14), heat impact (Fig. 15), and touch by a human (Fig. 16).

Interestingly, the tissue impedance of different plant species is different. Fig. 17 shows RMS impedance and Nyquist plot responses from Dracena plant and from Cactae plant tissues. The difference between two plant species is not easily spotted in the RMS impedance measurements but in the Nyquist plot difference it is evident. Based at such experiments we can conclude, that correct data representation and processing needs to be chosen to exploit the maximum numbers of feature which can be extracted from measurement data. The response of three different plant species (Dracena, Cactae, Ficus) for periodic stimulating signal shown in Fig. 18. The different types of sensors and plant reactions for different stimuli are shown in a video[11].

[11]plant sensor video available online: https://www.youtube.com/watch?v=e44dds1qvGM

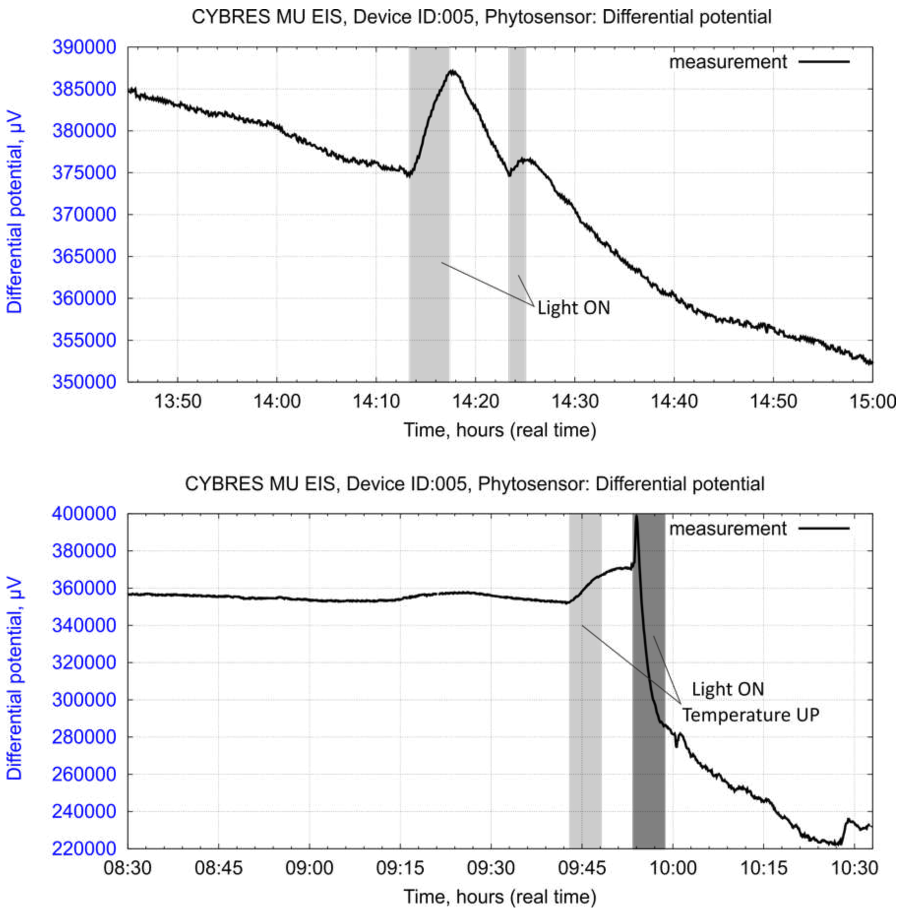


Figure 14: The Dracena plant tissue impedance measurement reaction for light impact.

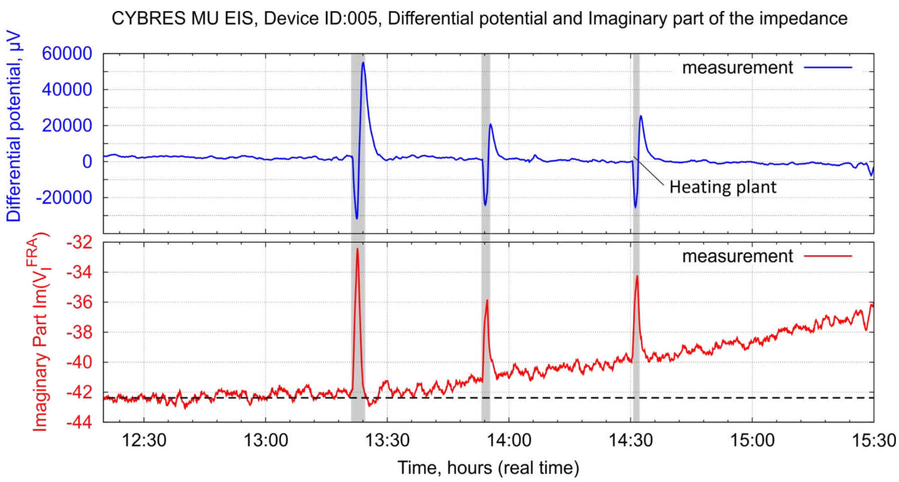


Figure 15: The Dracena plant tissue impedance measurement reaction for a heat impact.

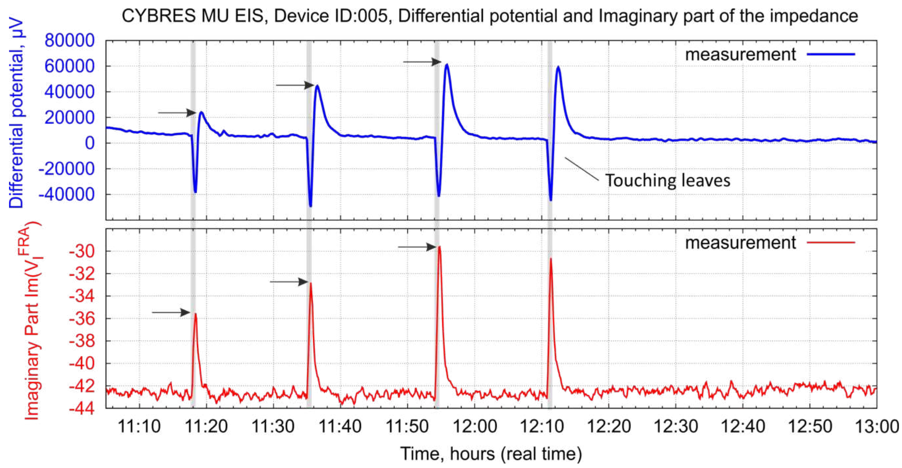


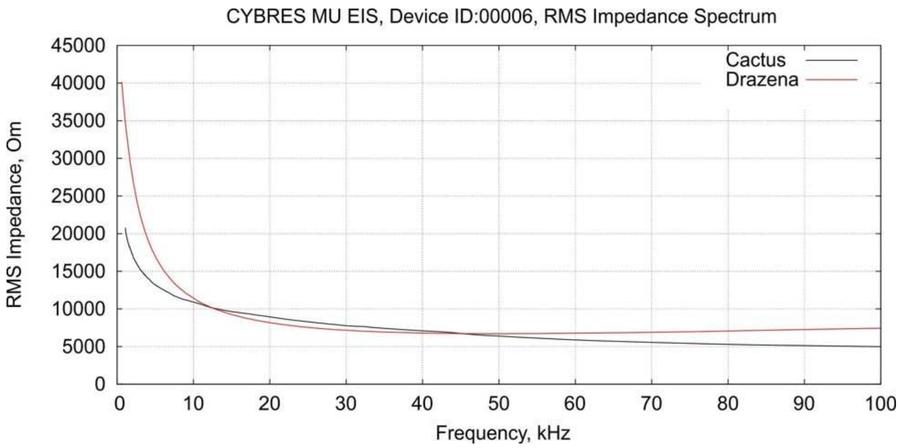
Figure 16: The Dracena plant tissue impedance measurement reaction for touch impact.

2.2.10. Long-term experiments for collecting electrophysiological data

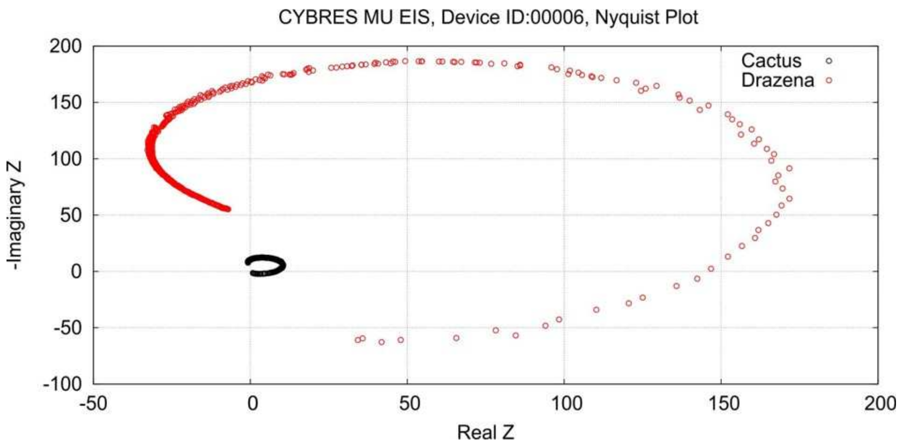
Four phythosensors (Dieffenbachia, Dracaena, two cacti) are connected to the Internet, the devices plot all measured electrophysiological data (two channels of biopotentials and tissue conductivity) in real time. Electrodes Ag925 d=0.25 are installed in the stem parts of plants (see Fig. 19). Along with Ag electrodes were used electrodes from another materials, such as stainless steel, copper, and nickel. Links to all sensors can be found at `http://cybertronica.de.com/OnlineMeasurements`, and see Fig. 20. On the page of a particular device, as shown in Fig. 21 the caption of the graphs indicates the device number, the recording time in the format "month:day:hour:minute:second", and the graph name. Not only electrophysiological data, but also environmental parameters are measured and plotted, among them: RF power 50Mhz-3Ghz, air temperature (and air humidity), PCB temperature, environmental light, 3D magnetic field, 3D acceleromenter, see Fig. 22.

All data is measured and plotted in real time, the interface also allows connecting actuators (RGB light, temperature and humidity) for performing online experiments. This service started September 2016 and runs continuously since then (except for short maintenance interruptions).

In summary, the developed electrophysiological system allows to accurately measure biopotentials in plant tissues and to measure the plant tissue impedance/conductivity using the method of EIS (Electrochemical Impedance Spectroscopy) with AC voltage at frequencies of up to 500 kHz. The system has two channels for each measurement technique allowing to measure, for example, bio-potentials and impedance at two different branches of a plant. These more sophisticated sensors will extend the capabilities of the PBDW robotic nodes and will help to create better interfaces to natural plants.

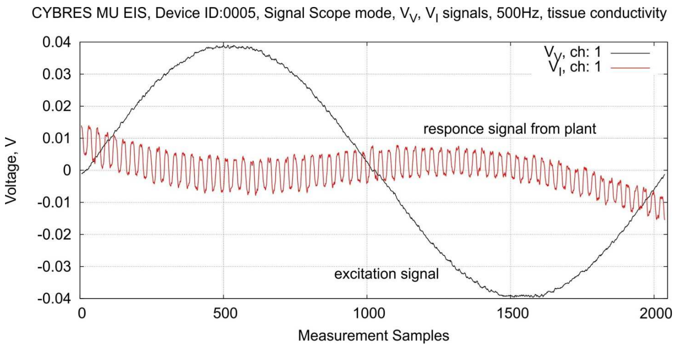


(a) RMS Impedance of plant tissues

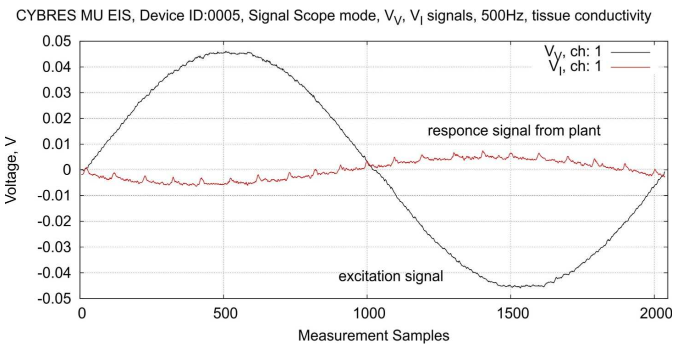


(b) Nyquist plot of plant tissues

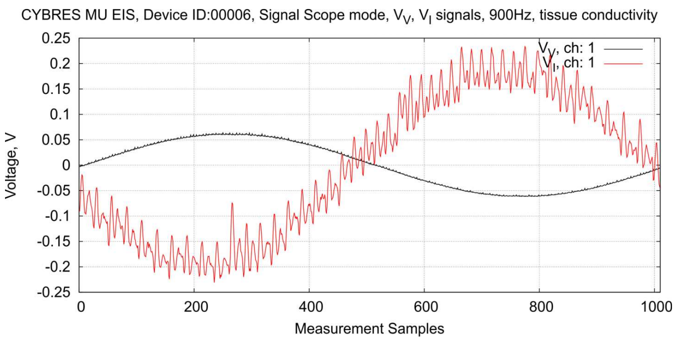
Figure 17: The RMS Impedance and Nyquist plot of Dracena and Cactae plant tissues.



(a) Dracena

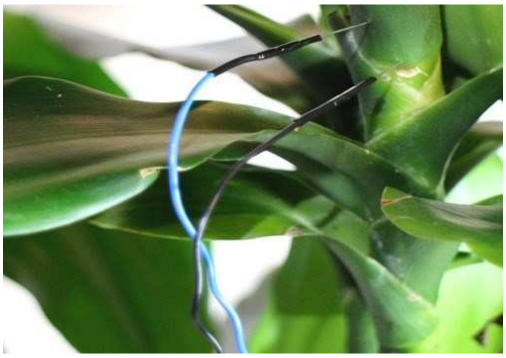


(b) Cactae

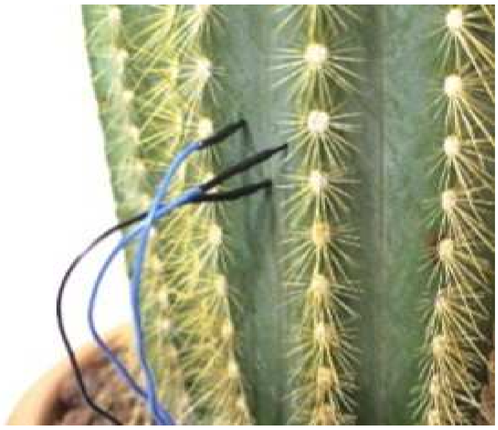


(c) Ficus

Figure 18: Different species plant reaction for 500 Hz periodic stimulating signal.



(a) Dracena plant



(b) Cactae plant

Figure 19: Needle-type electrodes injected into different plant plant tissues.

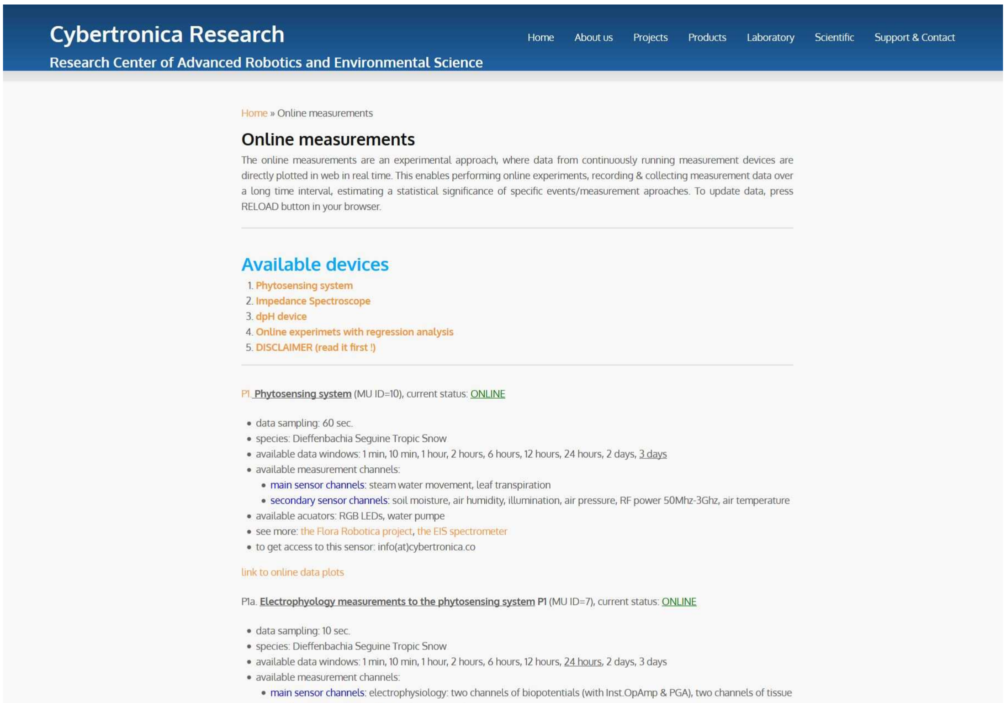


Figure 20: Main page for online measurements.

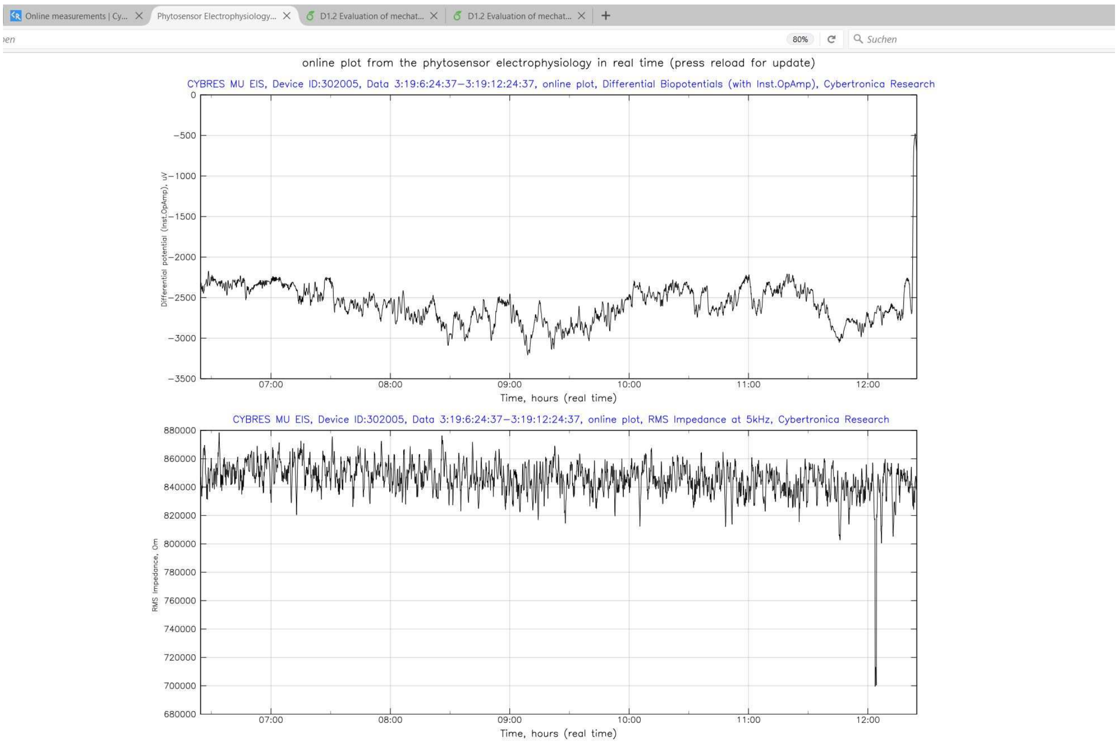


Figure 21: Example of web data from a particular electrophysiological sensor (image from cybertronica.de.com/OnlineMeasurements).

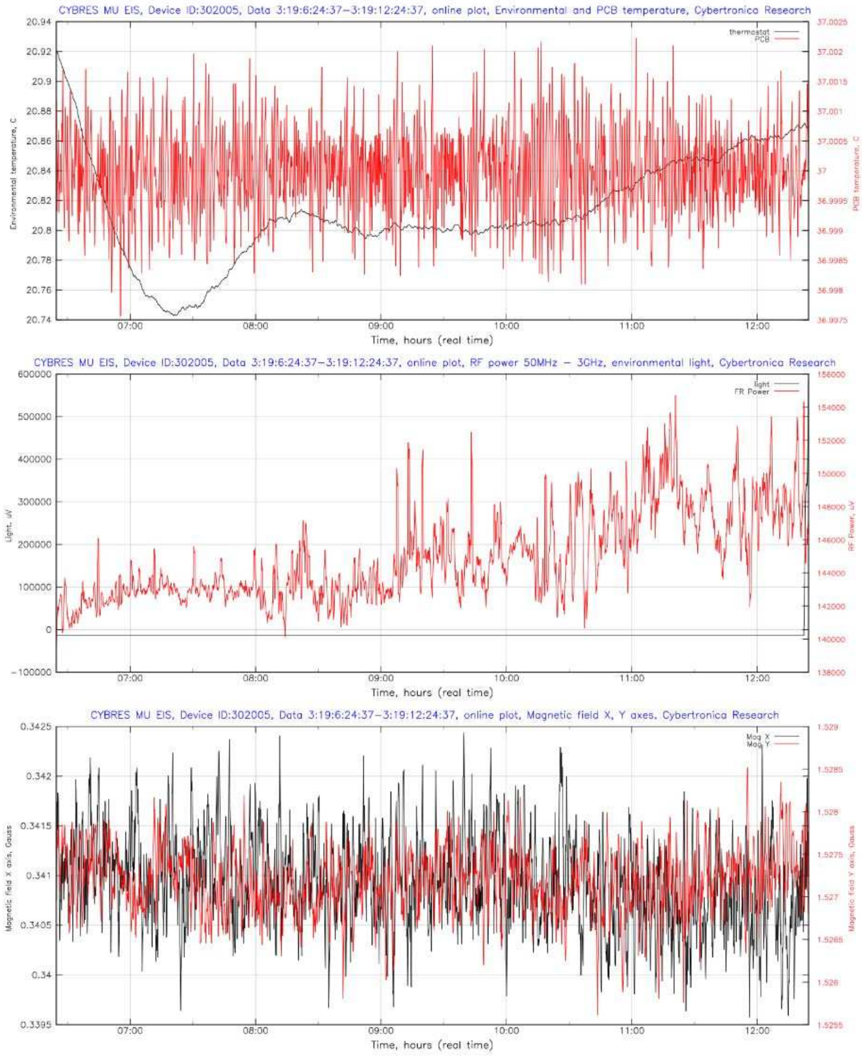


Figure 22: Example of web data from several environmental parameters (image from cybertronica.de.com/OnlineMeasurements).

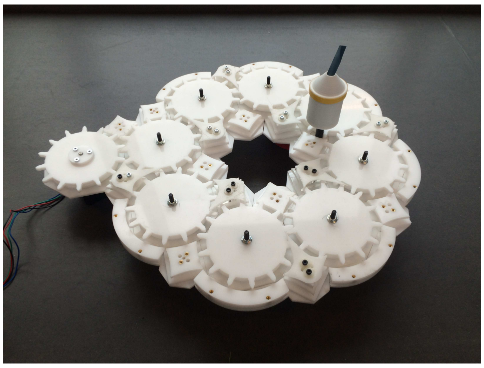
2.3 Production of topologically differentiated braids

A reconfigurable modular braiding robot has been developed that allows for production of multiple novel braid topologies. The diversity of braiding topologies is owed to the flexibility of reconfiguration and the programmability. We demonstrate different braided topologies and discuss how the high and low-level software can be improved for handling high-level braided definitions.

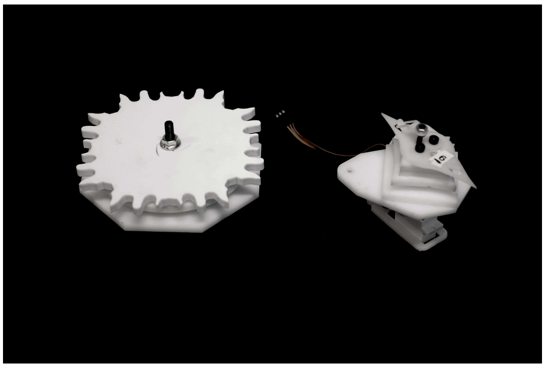
Modular robotics and swarm robotics have been the basis of our inspiration for developing the machine elements. One aim of the mechatronic device is to expand on the diversity of braids that can be produced by making the overall robotic braider modular and re-configurable in line with existing research [3, 2, 6]
Although our implementation is similar to some of the related work that is also modular, our approach is in addition able to (1) reconfigure the machine to produce various braiding topologies, (2) control the motion of the braiding reels locally, and (3) run the braider continuously. These advantages are verified through different configurations of the modular system which demonstrate the versatility of modules in combination with control by producing topologically different braids.

2.3.1. Design

The braiding modules are organized in an octagonal pattern giving each braiding module up to four neighbor modules. In the center of each module is a gear with specially cut teeth able to move material carriers around. These carriers contain the filament material wound up on a reel. The gears of neighbor modules rotate in opposite directions. At any of the eight sides of a module either a servo-actuated switch or another module can be mounted. The switching module controls the path of a carrier and can thus determine whether a carrier is switched to a neighbor module.
Reconfigurable: The modular design allows us to layout a braiding robot in a specific configuration. For example, modules laid out in a row perform a simple flat braid, circular configurations perform tubular braids, and matrices of modules perform solid braids.
Hybrid actuation: While each module in theory should be actuated locally by its own actuator in order to ensure scalability of the system, we found that a mixed approach, where motors can easily be added to elements proved both robust and scalable. In the small and medium sized systems shown here it was practical to actuate all modules using one to two actuators randomly placed.
Actuation of switching gates: In our original design, the switches were actuated with the passing of a carrier. This provided a simple system for producing simple, static braids. However, in the current system the switches are controlled individually by a servo motor allowing for complex, non-static braid patterns across layers. In the current system some switches are controlled mechanically and a few gates are controlled centrally. This already provides a significant degree of flexibility as we can deploy under-actuated switches in parts of the structure where the braid is expected to be static and use fully actuated gates where the braid is dynamic.
Control: In order to make a specific braid we need to control the way carriers move around the system globally by controlling the switching gates locally. This is a cumbersome task computationally as the global position changes for every local position change.
Material for braiding: For material we use gift ribbon and package straps as this makes it possible for the carriers to contain significant amounts of material while the braided structure can be almost a continuous surface.



(a)



(b)

Figure 23: (a) Configuration of 8 driver modules, under-actuated switches and one motorized module. One carrier is in the machine. (b) Two types of modules, driver and switch.

2.3.2. Implementation

In the process of braiding many continuous filaments are interlaced through a structured pattern. If the topology of the braid is altered the pattern of interlacing must be changed. The continuous filaments – of a given material – are dispensed from a number of carriers that are transported with the predefined pattern. The main contribution is the design and functionality of the robotic modules that facilitate the programmable movement of carriers.
The overall interlacing pattern can be defined partly by the configuration layout of modules. This layout of modules therefore defines the possible extend of produced braid topologies, however the specific pattern and production of braid is defined through logic control of the switches in coherence with the initial location and number of the carriers.

2.3.3. Mechanical implementation

Two main types of modules are designed. One type for driving the fiber carrier ‘driver module’, another for ordering the carriers passing by ‘switching module’.
The driver module is octagonal on its connecting sides such that it conjointly on a flat surface leaving space for the square switching modules. The octagonal shape also allow for 45-degree connections between modules that prove useful at circular configurations for braiding tubes. This capacity for different formations is what makes this braiding machine versatile.
The square switching modules tie together all modules by connecting to the driving modules but more importantly, they direct the stream of carriers by a servo actuated rail-switch. While the driver modules propel the carriers through the modules, the switching modules direct the path of the individual carrier, such that the intended pattern forms in the filament.

The switching module has two wings extending under the drive modules. This allows to connect to two drive modules at a time, making any larger configuration structurally stable and extendable. The drive module has 8 threaded holes to which switching modules, feet and additional motors can be attached.
When drive modules and switching modules are connected they create a concave track running in a circle around each drive module. This track holds the disk-shaped foot of the carriers standing

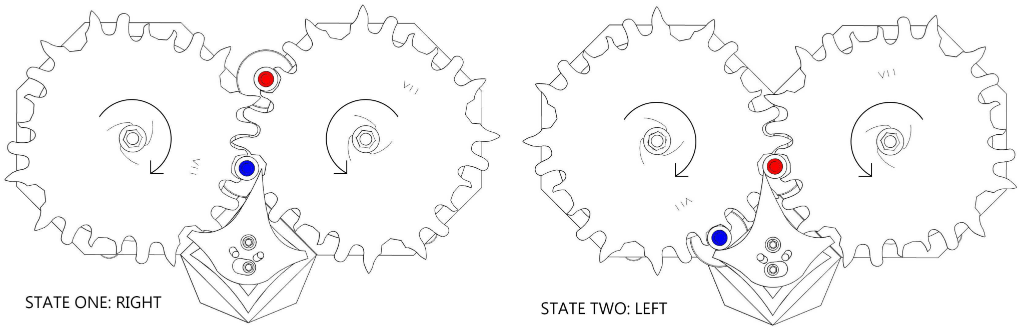


Figure 24: Under actuated switch changed by passing carrier transported by driver wheel

and prevent it from being pulled out.

The drive module has a set of ball bearings to hold the main driving wheel rotating on the top (see Fig. 25). These wheels are interconnected to adjacent modules with specially designed cogwheel teeth. The design of the teeth has integrated 8 slots for the carriers to be held while transported throughout modules – the slot acts as the slot in a Geneva Drive. The motion of the driving wheel propagates to all modules, and thus reduces the need for individual motors in each module. The system has friction and is prone to jamming, therefore an optical encoder is connected to one module, acting as main feed-back for the central micro-controller.

2.3.4. Low level control

The signal from this encoder and driver wheel is used for global positioning as well as to find the exact moment for shifting the switches. Using an encoder here allows for very accurate timing of the switch actuation (time that it takes the switch to turn 20 degrees is required for a safe switching operation).

One layer of software control is handled at the micro-controller level, while another is running as an application on a laptop. The micro-controller is given the program code to execute, while the application is guiding an operator in loading the carriers onto the driver modules in the correct order. The same application is used for verifying the route of the individual carriers and for checking that no collision occurs. The two layers exchange information on how the rotary wheels are positioned and to which positions they should go. Due to the need for high temporal accuracy of the servos in relation to the drivers' position, the program that is managing the order of switching must be buffered on the micro-processor.

2.3.5. Simulation application and machine programming

The simulation model consist of mainly two classes of objects; carriers and switches. Both types of objects define their virtual location by the module and position in each module (i.e., positions 1-8 given by the edges of the module). The switches are similarly defined by the two modules it switches between, as well as the position at which the switch is placed in the given configuration. The position is mirrored over the connecting edge between all modules. This also means every second module turns clockwise and the rest counterclockwise. This, in turn, means positions always align where the drive wheel touches. The position of all carriers is incrementally increased as the simulation plays out. Once a carrier shares its position with a switch, the carrier queries the switch for its connecting modules and whether it is turned left or right. The carrier then

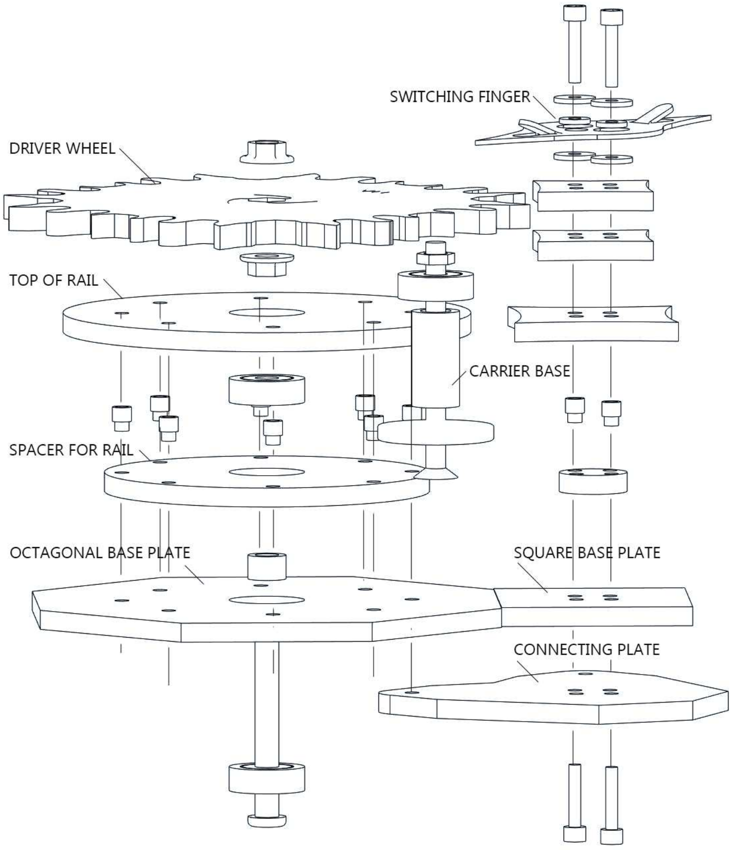


Figure 25: Exploded view, showing the layered built of the two modules and the carrier in the middle

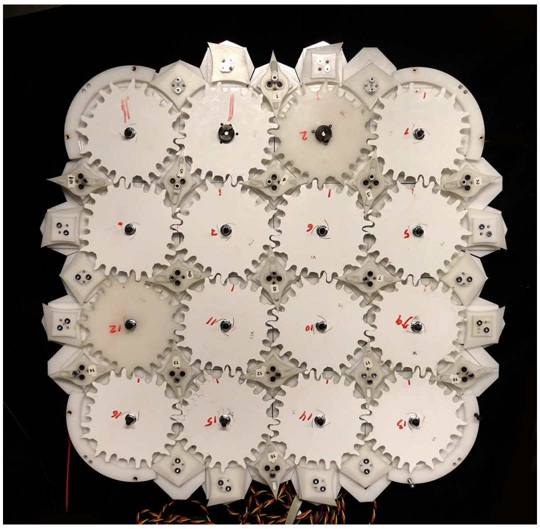


Figure 26: Configuration of 4x4 driver modules. The switches off the edges are acting to both sides, managing 4 modules each.

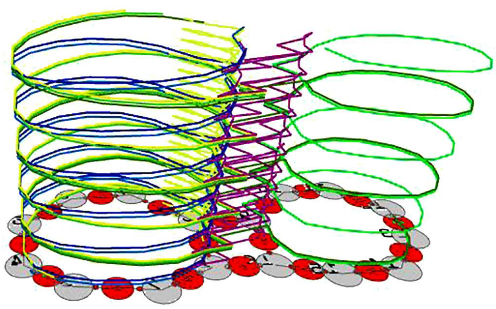


Figure 27: Software application simulate a ‘double tube’ program. For simplified graphics only one module is populated with 8 carriers at initial position.

changes modules or stays in its current module. For the switches to have the correct state when a carrier arrives, a program can be sent to the micro-controller. A string of '0' and '1' indicates which switch is set to left or right in the given position. A simple program as used in the experiment below would thus look like:
"1100100"
"1100011"
"0011111"
"0011000"
"1100100"
"1100011"
"0011111"
"0011000"

In this case the program runs this sequence in a loop until stopped externally.

To search for collisions each carrier location is compared to all other carrier location. Collisions have proven an easy way to test new configurations and programs. All slots are occupied initially and the simulation tool then sifts out one element at every collision. In turn a solution is found where a number of carriers fit the given configuration and program, however the result may not be useful. In simulation we are able to calculate the sequence of switching for a given start and end location of a number of carriers. This calculation may however become very large and may not have a solution. There is still much work to be done finding a way to create optimal programs.

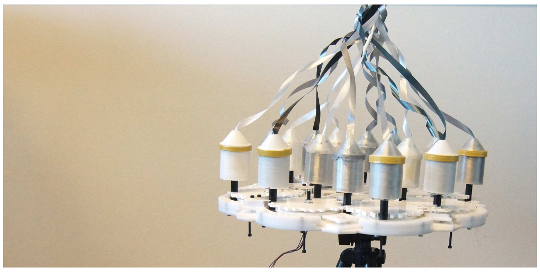
In turn the model is visualized by a graphical user interface. In this 3D environment each module can be drag-dropped into position to match the physical layout of a given configuration. Insofar, switches in the model must be encoded to match position in the physical configuration. This 3D graphical layout of modules visualize x, y, z positions of each carrier and while the sequence of the program plays out, the trace of their individual route is visualized. The result is a 3D representation of the extruded path of filament. The data from this representation could in the future be useful at dynamic relaxation or FE-analysis of the unbuilt braid. With the simulation tool we can test different programs and visualize how initial placement of carriers affect the outcome.

2.3.6. Experiments

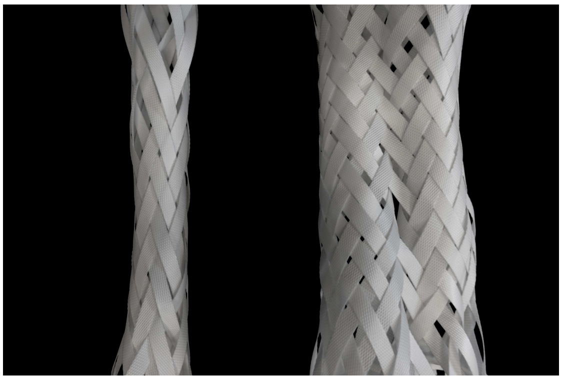
Three different configurations of the modules have been tested to explore the variation of braid topologies possible with reconfigurable modules. The software for creating programs for more advanced configurations is yet to be developed.

Configuration of 8 driver modules in a circle with 8 under-actuated switches and 16 carriers: This configuration is controlled only by the passing of carriers and by the speed of the revolution (see Fig. 24). Initial position of the carriers were 4 in every second driver module. This braid could be changed by switching all inner switches once during operation, resulting in a global change in the movement pattern thus producing 8 strands of two filaments instead of the single tube of 16 filaments. After switching the four switches over again and resuming operation, the braid goes back to producing one strand of a tube from all 16 filaments.

Configuration of 16 driver modules in two connected circles with 17 actively actuated switches and 32 carriers: Given the same initial position of the carriers – namely

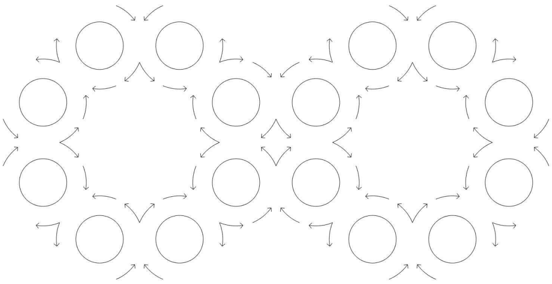


(a)

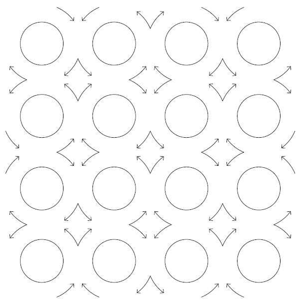


(b)

Figure 28: (a) 8 modules, 8 under actuated switches, 16 carriers in circular configuration, braiding a tubular braid. (b) Tubular braid. Right, shared double tube, some filaments are shared while some enclose the individual tubular braid.



(a)



(b)

Figure 29: (a) 16 modules in an 8-shaped configuration. (b) 16 modules in matrix configuration

four distributed in every second drive module – the switches can be programmed to produce a topology of two and one tube simultaneously (see Figs. 29(a) and 28(b)). Initially 4 carriers are placed in every second drive module. The switches along the outside of the conglomerate are grouped into two main groups in order to switch simultaneously. The three out of four inside switches also connect in two different groups. The last tree switches, two on the inside and the one centrally placed switch make up their own groups. This grouping removes control redundancy because if any of the members in these groups were turned independently it could lead to collisions, while being without function. The programming of switches guides some carriers through the two rings without crossing over, while others cross through the middle and only revolve around the one ring.
The program and execution results in a braid which topologically can be seen as a continuous double surface. A cross section looks like an 8 inside an O sharing top and bottom surface.

2.3.7. Discussion and further development

The configuration of 16 driver modules in a matrix with 15 actively actuated switching modules is not yet tested (see Fig. 29(b)). Ideally we will produce solid braids, where filaments will move in each row from left to right and top to bottom. The configuration is interesting because a solid braid is a fundamentally different topology from the more conventional surface-based braids. In future work, we will test the combination of these different topologies as well.

Software for handling the carriers routing through the modules may be solved successfully through the use of concepts from swarm and modular robotics. The constraints for this transportation are strict, but on the other hand it is a rather large state space to explore. We have yet to figure out how high-level braid descriptions are translated to switching commands, but a viable approach is to consider the carriers as agents with an intend of route or track. A high-level representation may be translated through a form of projection that lays out the the tracks. Secondly, it is an iteration task to find the initial position and assign tracks correctly to carriers. The carriers' capacity to dispense various fibrous materials is essential to the functionality. We found that flat materials provide interesting features for surfaces and tubular braids. Flat material, however, needs some control of directionality while being dispensed. Furthermore, control of tension and thereby braiding angle is important to many applications. The carrier should therefore be capable of dispensing both flat and round filament with accurate tension or even accurate length. These aspects will be addressed in later design iterations.

With this design, braiding technology can become a novel addition to fibrous deposition technologies and rapid prototyping. The design is extremely cost efficient and can be reproduced with the availability of a standard laser cutter or CNC milling machine and a standard FEM 3D-printer. All mechanical parts are either cut out of PMMA or POM plastic sheets or easily available for on-line purchase.

For *flora robotica* the capacity to manage and produce more complex patterns of braids is essential. This capacity may lead to more accurate braids of a higher complexity that will be fulfilling multiple performance aspects simultaneously.

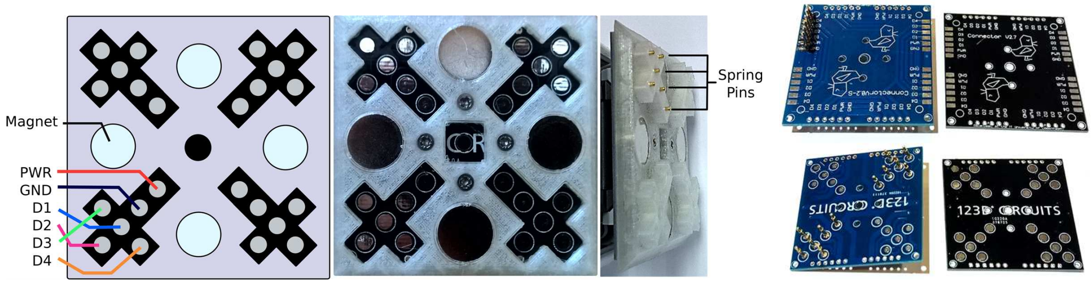
2.4 Exploring the potential of energy autonomy of a robotic symbiont

The previous sections have discussed the importance of braided structures in the mechatronic design of robotic elements in *flora robotica*. However, since a braid is hard to control, let alone simulate, this section explores how we can envision robotic elements working together for a specified goal. Instead of focusing on the novelty of the material, this section views the material merely as a functional part for a yet unspecified purpose. We developed a platform for combining various robotic elements to create unique robotic prototypes from modular building blocks. Though the current approach does not include features of the braided mechatronic principles (for simplicity), it does allow for future integration of braided structures in the system. This system can be viewed as the skeletal backbone that could shape the ultimate (probably non-modular) implementation of the bio-hybrid system.

2.4.1. Motivation on modularity

The important benefit of modularity in robotic systems is the quick evaluation of reconfigurable robotic systems in reality. A robotic module is an independent unit that encapsulates parts of its functionality [9]. Apart from the similarity of a modular approach to natural systems (considering e.g. self-similarity, reuse of genetic information, evolvability), this approach has some clear practical benefits for creating hardware components. The main mechanism conveying how the modules are attached to one another is depicted in Fig. 30. This connection mechanism allows for any type of module to be attached in 4 different orientations while still being able to connect power, ground and up to 4 data channels to the next module. All channels are shared and the servos of the entire robot can be addressed by using only one data channel.

Since the eventual hardware is somewhat unspecified, this modular system would allow for the integration of new robotic elements that might have a unique manner of actuating, sensing or deforming. Instead of needing a new power source or communication system, a new module with a specified functionality can simply be plugged into the system – be it a robotic actuation module, a humidity or infrared sensor system, or a deformable braided module. There are five advantages of using this modular system in the project: (1) it enables for a human in the loop approach; (2) it allows us to iterate through various independent robotic units without redesigning an entire new robotic entity; (3) redundant modules can simply be taken off and be reused elsewhere in the system; (4) it allows for a simple genotype to phenotype mapping in an evolutionary approach; and (5) modularity simplifies the mechanical design due to the reuse of similar parts. (1) Since modules can easily be plugged out of one part of the system and plugged into another part, we can interact with this system to adjust the performance of our robotic entity. Say a braided scaffold is misplaced, or simply restrains the motion of agents, we would be able to take off a piece of the scaffold and reuse it in a place where it would be more beneficial. In order to decrease the redundancy in our system, a modular approach would be beneficial for an extended version of the decision wall (discussed in Sec. 2.2). (2) If we need to actuate a specific part of the system we can simply add a servo module, even in between two already existing parts. Moreover, since braids are becoming more important in this project (see Secs. 1.2 and 2.3), a braided module can simply be integrated. If we designed a new sensor (or sensors defined in Sec. 2.2.8) for our system it would be easy to integrate by just latching on to one of the free attachment sites. No separate power source or data cables would be required since all channels are shared with all modules attached to the system. (3) This also allows for the convenient replacement of parts when a module breaks. (4) When evolving a system with unspecified hardware parts, the integration of new functionality through introducing a new module greatly simplifies the modeling of such a system. A simulator can simply state that it is using an additional module of a specific type. This type can range from functioning as structural



(a) (b)

Figure 30: (a) The connection mechanism of the modules is composed of a female connector (left and middle) and male connector (right). Each module contains six channels using which modules can distribute power (PWR), ground (GND) and data (D1, D2, D3, D4) from one module to others that are attached to it. Both the male connector and the female connector have copper pads. Spring pins are attached to the pads on the male PCB. These spring pins (right) allow for the information to be transmitted from one module to the next. The connection sites contain magnets through which male sites can be connected to female sites. (b) The PCBs used in the connection sites. the blue (left) PCBs have places to connect the spring pins and is used for the male connector sites while. The black (right) PCBs are used for the female connector sites.

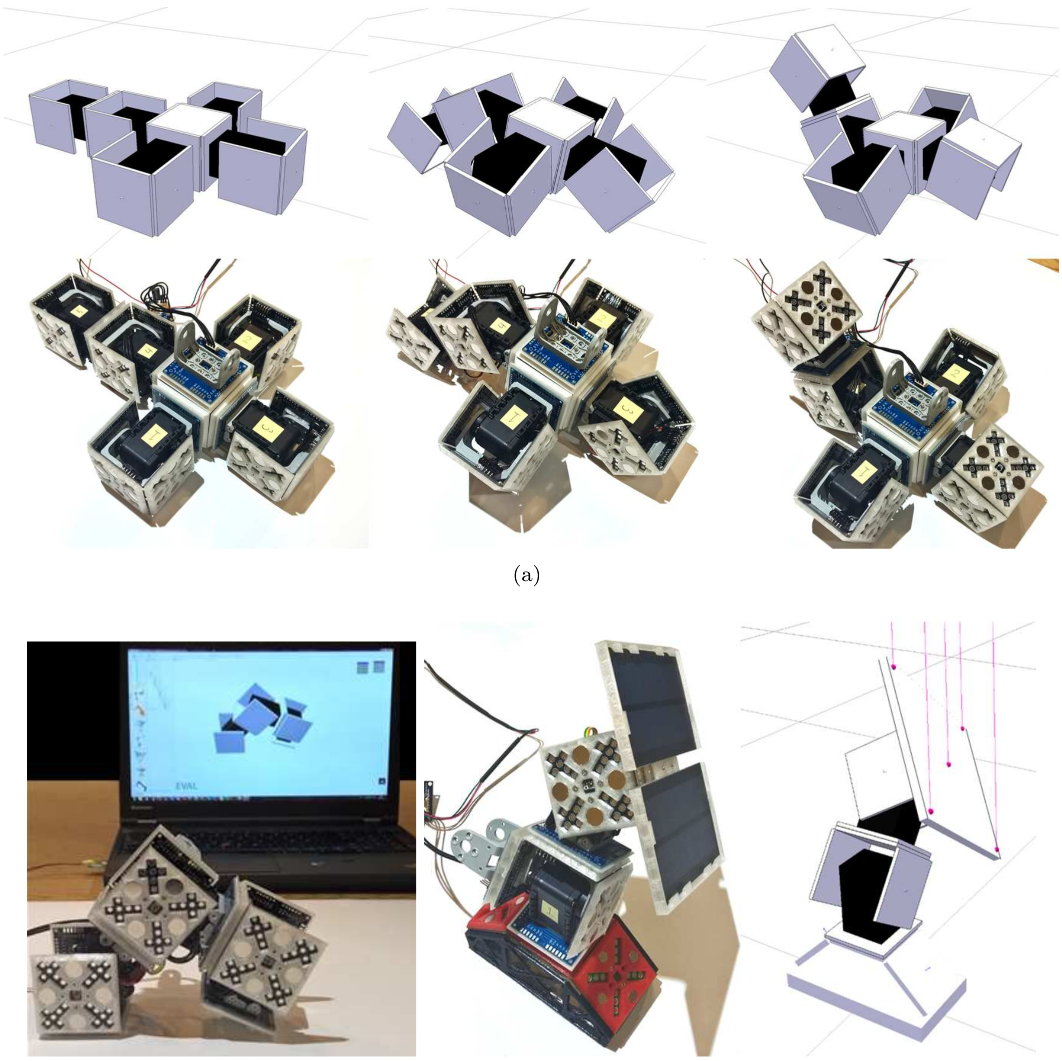
support to sensing the environment. One would only need to design the core functionality of the module while the simulator integrates the rest. Depending on the goal of the robotic system, the simulator would automatically integrate the new module and allow for unique robotic systems to evolve. (5) Simulating a mechanical structure, especially in the light of emergence, is tedious. Modularity can help to simplify the construction process of both simulated and real robots. For example, many modules have the same parts meaning that only a few designs of 3d printed parts are necessary. For more information about the evolution of modularity, please refer to [11] and refer to [10] for the relevance to *flora robotica*.

2.4.2. Robotics platform

The designed robotics platform consists of two parts: the simulator and the robotic modules. The simulator utilizes an evolutionary algorithm and a generative encoding to generate robotic entities composed of modular building blocks. These building blocks have physical equivalents that can be directly assembled in reality. The main elements that can be investigated with this platform are related to: the reality gap, the genotype to phenotype mapping, evolving morphology and control, embodied evolution, and resource management.

Simulator

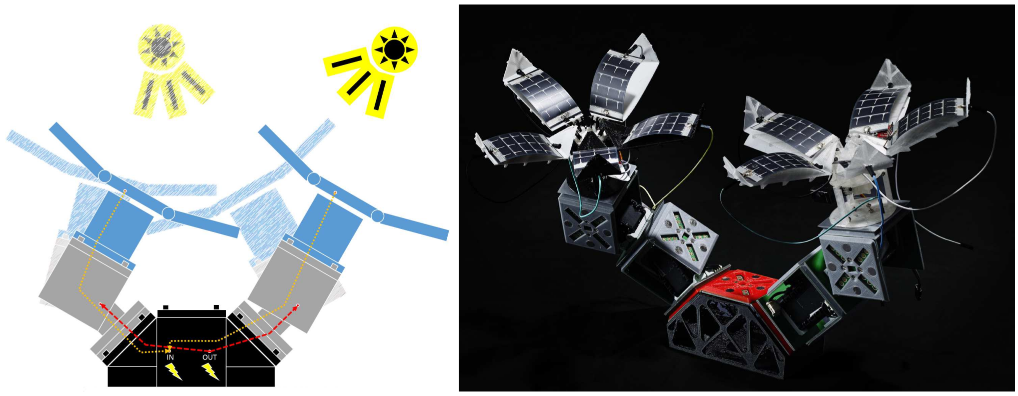
The modular approach is directly connected to the robotics simulator Virtual Robot Experimentation Platform (V-REP) [8]. A DLL plugin implements an evolutionary algorithm that can control, construct, and evaluate modular robots. Both a generative and a direct encoding is used to evolve robotic structures. The generative encoding is based on an L-System and has been shown to quickly lead to useful robotic entities. However, as in any evolutionary robotics setup, a difficult problem concerns the crossing of the reality gap. The simulator is set up to be directly able to connect to the modular robot. This enable us to quickly evaluate evolved robots



(a)

(b) (c)

Figure 31: Illustration of the modular robot being directly connected to the simulator: (a) The simulated robot with its corresponding physical state below; (b) 3 modules connected to the simulator; (c) a robot composed of two servo modules, one base module, and a solar panel module.



(a) (b)

Figure 32: Principle design of the Modular Solar Panel Robot: (a) a schematic overview of the design principle; (b) the physical robot

in reality. Especially in cases where our robotic system is required to be sessile, the reality gap can be easily minimized since we do not have to take into account complex collision dynamics that are usually a limiting factor in robots that are evolved for locomotion.

Robotic modules

The modules are connected to each other via the connection sites depicted in Fig. 30. All modules share the same power and communication channels. The modules transmit power ground and up to 4 data types through their connection sites. This thus allows modules to communicate with a central controller without the need of any additional wires or power sources when the structure is being reconfigured. The result of communication between the modules and the simulator is depicted in Fig. 31. Using the dynamixel software development kit, the plugin described in the previous section is directly coupled to the real robot. This enables us to test out the evolved models of robots to be directly tested in reality. The construction of the robots is however not automated. Although it is out of scope of the project, the evolved designs could potentially be extended with having a mechanism that automatically disconnects and reassembles the modules – either through a robot arm or an inherent mechanism of the modules. Without an automized construction progress we can keep the human in the loop. The aim of the modular robotic system is to automatically design abstract robotic systems that have a basic purpose for interaction with their environment.

As can be seen in Figs. 31(c) and 32, two robotic conglomerates have additional solar panel modules. Similar to plants, this is the first step of a modular robotic system towards online energy autonomy. In our approach, since the current flowing through the solar panels is relatively small, two data channels are used to have a current flow in opposite direction with respect to the main power source. This enables a power source to be automatically charged by the solar panels. Although Fig. 32 depicts a complex implementation of a solar panel module, this module can potentially be extended with an mechanism that is similar to origami folding techniques that enables the potential surface area of the module to increase dramatically.

Experimental relevance

Various approaches can be undertaken for designing a robotic system. This section seeks to explore questions related to the reality gap as well as the emergence of robotic systems. Since *flora robotica* also has a biomimetic aspect to it, exploring how an artificial ecosystem similar to nature can be envisioned is also part of the project. This can eventually give us more insights in how to design a system with its symbiotic counterparts.

2.4.3. Advances, limitations and future prospects

The described robotic platform is ideally suitable for doing experiments on the reality gap since the simulator is directly connected to a robotic entity. The unique feature of this platform is that it can allow for morphological change, either in epigenetic, ontogenetic, or phylogenetic timescales. The robotic platform is furthermore able to integrate new modules as long as they utilize the same communication protocol. An eventual robot composed from these modules will only need a single power source, a single control system, and can potentially become energy autonomous with the integration of solar panels. In the case that we need our robotic system to provide new functionality, the design and integration of a new module should be easy and straightforward to implement. The magnetic connection system turned out to be very robust and is able to couple a set of 9 modules in sequence without hardware failure when lifting it in the air from one outer module. This however also means that you need to use some force to disconnect certain parts.

Some of the limitations of the modular approach includes the robot becoming heavy. This is a inherent problem due to the weight and redundancy of some connection sites. However, unused connection sites could still be disconnected through manually removing the screws. Another limitation is the large power consumption of the servo motors in the modules. This makes it harder to design an energy autonomous system but this also highlights the importance for robotic systems to become more energy efficient and the potential benefits for designing energy autonomous robots in the future. The system has not integrated any braided mechanics yet mainly due to the late onset of converging on braids in the *flora robotica* project as well as the complexity of designing a braid – especially when considering available structural alternatives. Deformation, structural strength, and elasticity can be implemented using alternative types of structures though the main advantage of the braid would be the direct integration of the plants in the physical structure (plant grows through the braid).

2.4.4. Concluding remarks

The modular robotic platform demonstrates its usefulness for investigating the reality gap as well as emergence in robotic systems. It is set up so it can easily be extended with additional functionality and it is integrable with other modules as long as the same communication protocol is used. New modules, be it sensors, actuators or structural units, can easily be integrated in the system. Furthermore, the goal of the simulator can be changed on the spot which can ultimately lead to new and unique robotic entities as well as enabling the investigation of, for example, incremental evolution, multi-objective approaches and online learning.

3 Conclusion

This concludes the overview of our eco-system of braid technologies. From the highly explorative first period of the project we have now, by the half-way point, established four pillars that will be the mechatronic basis for *flora robotica*. Namely, (1) braided robots and intelligent filament, (2) sensor-actuator nodes to interface a braided symbiont to the plant symbionts, (3) technologies related to the automatic production of braided structures, and (4) technologies exploring the potential of energy autonomy of a robotic symbiont. All strands are established to a degree that they are relevant to the broader scientific community as evident from the resulting scientific papers. However, more importantly these pillars are evaluated to be appropriate for realizing the vision of braid-based plant-robot hybrid systems providing architectural functions. Each individual mechatronic prototype investigation covers individual parts of the overall vision and the consortium will, in future work, combine discoveries from both representational and algorithmic investigations with the mechatronic prototypes introduced here. Furthermore, many synergies can be found between the mechatronic approaches.

References

[1] Ronald C. Arkin and Magnus Egerstedt. Temporal heterogeneity and the value of slowness in robotic systems. In *2015 IEEE International Conference on Robotics and Biomimetics, ROBIO 2015, Zhuhai, China, December 6-9, 2015*, pages 1000–1005. IEEE, 2015. ISBN 978-1-4673-9675-2. doi: 10.1109/ROBIO.2015.7418902.

[2] Kadir Bilisik and Nesrin Sahbaz. Structure-unit cell-based approach on three-dimensional representative braided preforms from four-step braiding: Experimental determination of effects of structure-process parameters on predetermined yarn path. *Textile Research Journal*, 82(3):220–241, 2012.

[3] Alexander Bogdanovich and Dmitri Mungalov. Recent advancements in manufacturing 3d braided preforms and composites. *Proc ACUN-4 Composite systems-macro composites, micro composites, nanocomposites. Sydney, Australia: University of New South Wales*, 2002.

[4] Sander W. Hogewoning, Emilie Wientjes, Peter Douwstra, Govert Trouwborst, Wim van Ieperen, Roberta Croce, and Jeremy Harbinson. Photosynthetic Quantum Yield Dynamics: From Photosystems to Leaves. *The Plant Cell*, 24(5):1921–1935, 2012. ISSN 1040-4651. doi: 10.1105/tpc.112.097972.

[5] Keith J. McCree. The action spectrum, absorptance and quantum yield of photosynthesis in crop plants. *Agricultural Meteorology*, 9(C):191–216, 1971. ISSN 00021571. doi: 10.1016/0002-1571(71)90022-7.

[6] Dmitri Mungalov and Alex Bogdanovich. Automated 3-d braiding machine and method, August 27 2002. US Patent 6,439,096.

[7] Theoharis Ouzounis, Eva Rosenqvist, and Carl-Otto Ottosen. Spectral effects of artificial light on plant physiology and secondary metabolism: A review. *HortScience*, 50 (8):1128–1135, aug 2015. URL http://hortsci.ashspublications.org/content/50/8/1128.abstract.

[8] E. Rohmer, S. P. N. Singh, and M. Freese. V-REP: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326, Nov 2013. doi: 10.1109/IROS.2013.6696520.

[9] Kasper Støy, D. Brandt, and D.J. Christensen. *Self-reconfigurable Robots: An Introduction*. Intelligent robotics and autonomous agents. MIT Press, 2010. ISBN 9780262013710. URL https://books.google.com/books?id=WhhFAQAAIAAJ.

[10] Frank Veenstra, Andres Faina, Kasper Støy, and Sebastian Risi. Generating artificial plant morphologies for function and aesthetics through evolving L-systems. In *Proceedings of the 14 Int. Conf. on the Synthesis and Simulation of Living Systems (Alife)*, pages 692–699. MIT Press, 2016.

[11] Frank Veenstra, Andres Faina, Sebastian Risi, and Kasper Støy. Evolution and morphogenesis of simulated modular robots: A comparison between a direct and generative encoding. In *20th International Conference on the Applications of Evolutionary Computation, Evostar 2017*, volume 20, 2017. (in press).

[12] Mostafa Wahby, Daniel Nicolas Hofstadler, Mary Katherine Heinrich, Ewald Neufeld, Igor Kuksin, Payam Zahadat, Thomas Schmickl, Phil Ayres, and Heiko Hamann. Shaping natural plants by distributed robots: Towards bio-hybrids for construction. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2017. (submitted).

A Appendix. Application Note for Raspberry Pi Hat.

Application Note 16. Electronic design of D2 node board to interface the sensors and actuators to Raspberry Pi in the *flora***robotica** project

1. General description

The D2 node board developed specially to D2 decision wall demonstrator in *flora***robotica** project. The board act as interface module between Raspberry Pi single board computer with sensors and actuators. The board have the same size Raspberry Pi Zero – 65 x 30 mm. The D2 board connecting as a hat to Raspberry Pi with 40 pin GPIO header connector. The D2 node board compatible to use with several Raspberry Pi models: Zero, 1 model A+, 1 model B+, 2 model B and 3 model B as well as these boards have the same 40 pins connector configuration.

The board have 8 channel 10-bit SAR ADC chip with SPI interface. The board have 2.54 pins headers connectors designed to acquire the data from the **4** LDR sensors (light depended resistors) and **3** IR (infrared) proximity sensors supposed to be connected to this pins headers. The pull-up resistors to LDR provided at the board as well. The 4.5 V 250 mA ultralow noise switchable LDO provided to power all analog sensors and periphery. The ADC have 4.096 V precise voltage reference and analog inputs op amp buffered to perform accurate analog measurements. The measurements from the sensor will be acquired via SPI interface, provided by ADC chip. Also, board have connections to interface the DS18B20 temperature sensor to Raspberry Pi via 1-wire protocol.

From actuator side board have two channel general purpose n-MOSFET. It can be used to control the load with 5V DC voltage with current up to 30 A[1]. The power 3.5 mm screw 5V 10A [2] terminal placed at the board to connect the Pixie LEDs. The Pixie have its own controller with UART (Universal asynchronous receiver/transmitter) interface. So, Raspberry Pi will control Pixie LED via UART. The LLC (logic level converter) from 3.3 V Raspberry Pi logic to 5V Pixie LED logic provided at the board.

Features

- 8 channel 10-bit SAR ADC with 4.096 V precise voltage reference and buffered analog inputs
- 4 LDR sensors
- 3 IR proximity sensors
- Temperature sensor with 1-wire interface
- 3.3V to 5V LLC for UART
- 4.5V 250 mA ultralow noise switchable LDO
- Power 3.5 mm screw 5V 10A [2] terminal (16 AWG wire diameter max) for Pixie LEDs
- Board dimensions is 30 x 65 mm

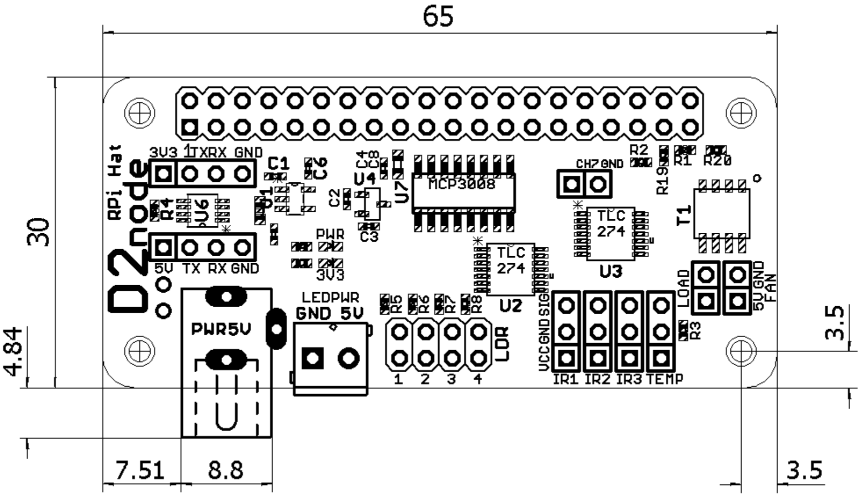
[1] 30 A – the limitation of n-MOSFET. The whole system current limitation depend from power supply limitation, the connectors as well.
[2] 10 A – the limitation of power connector and board traces. The whole system current limitation depend from power supply limitation as well.

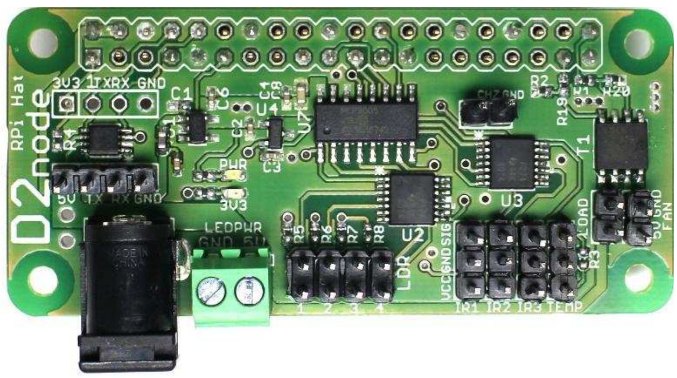
2. Versions

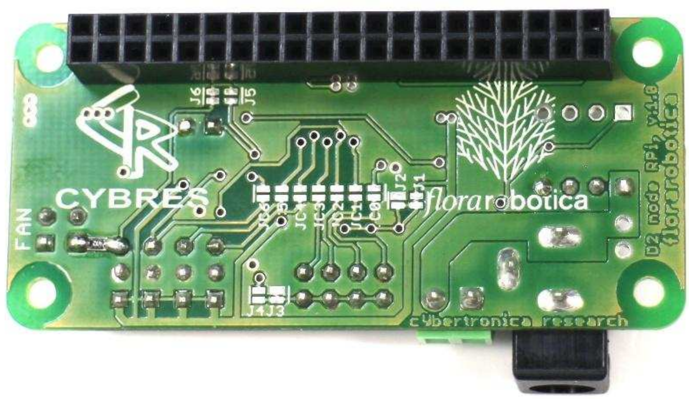
Version 1.0

- board size: 30 x 65 mm
- production: May 2016

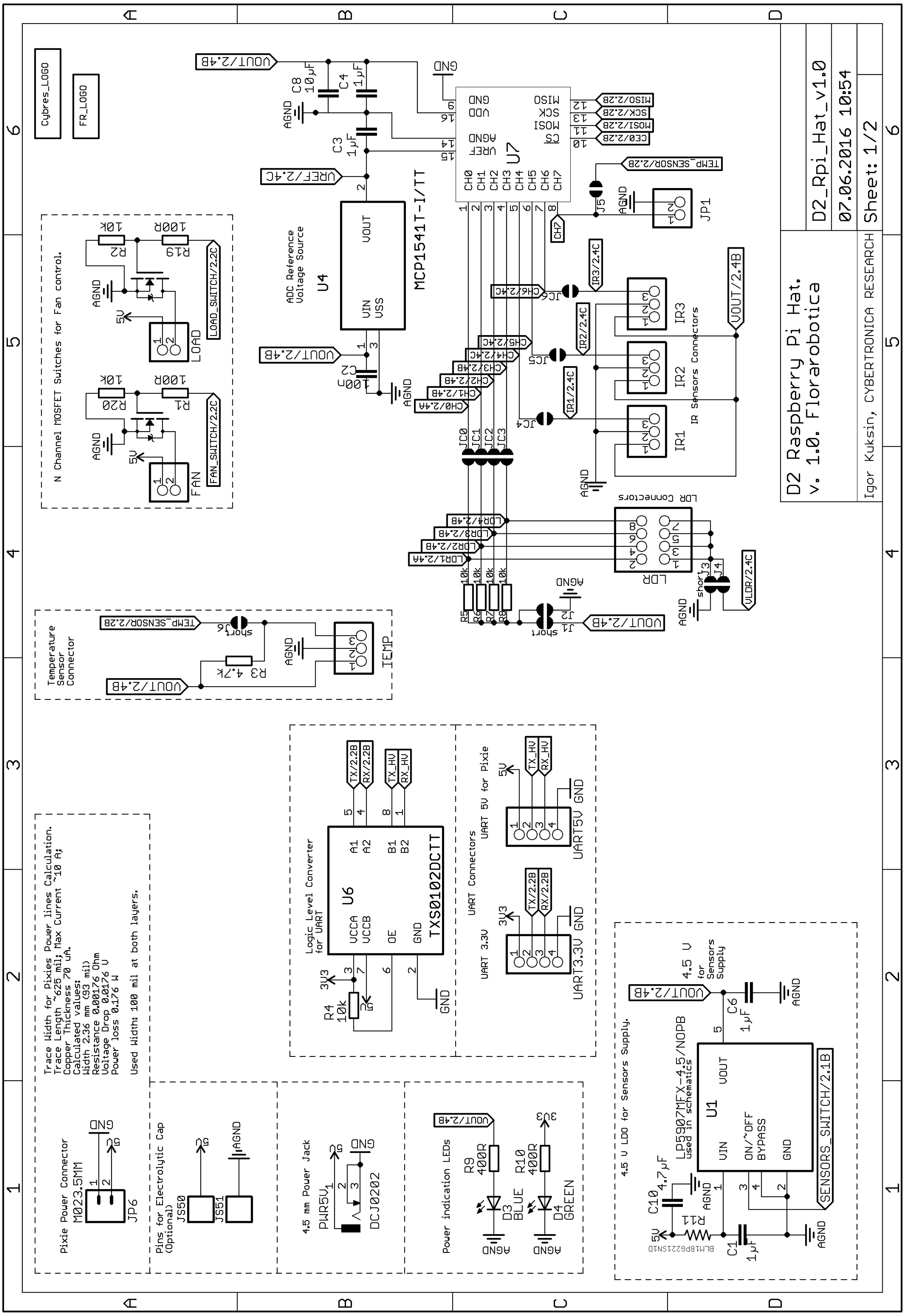
3. The board outline dimensions

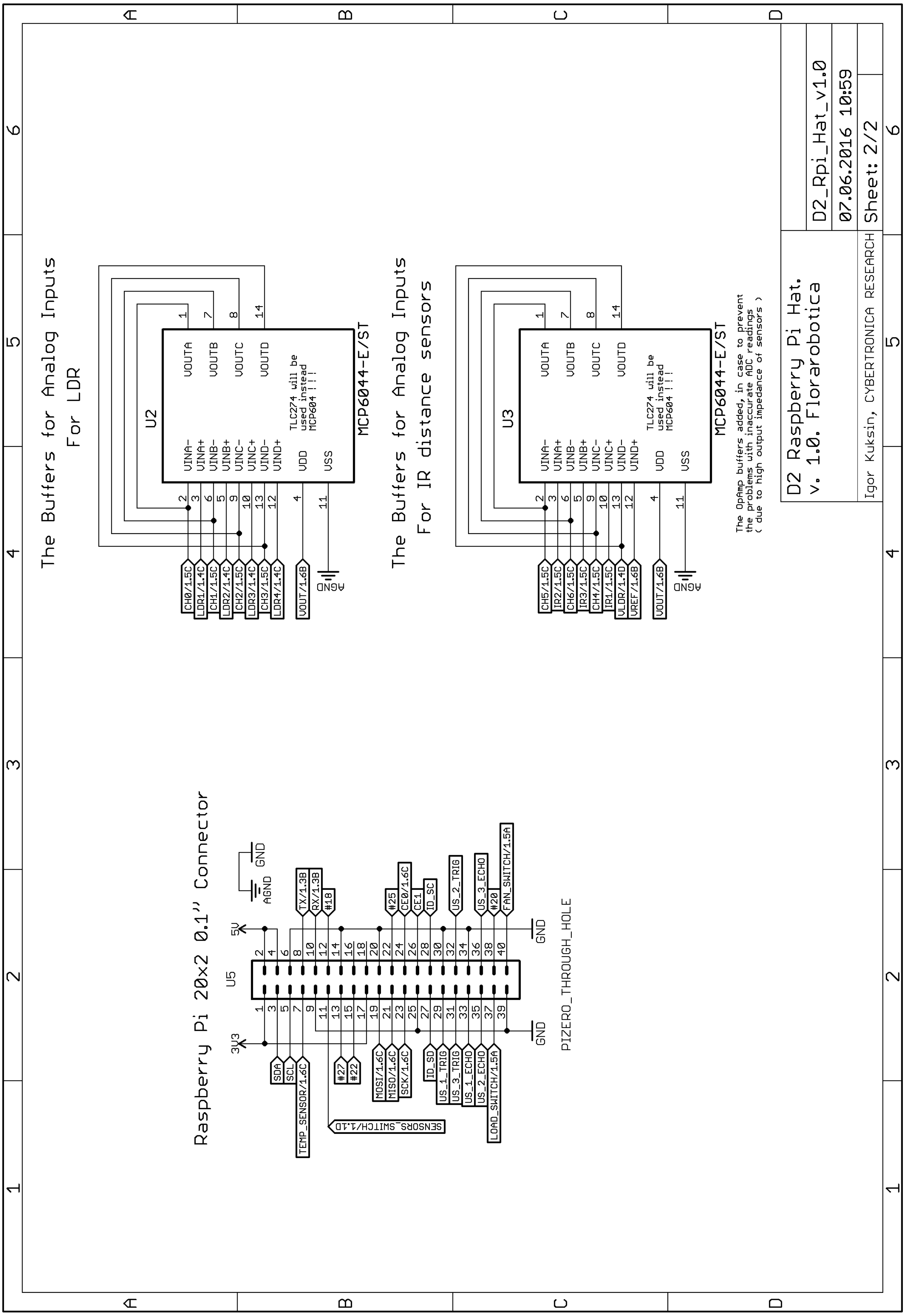






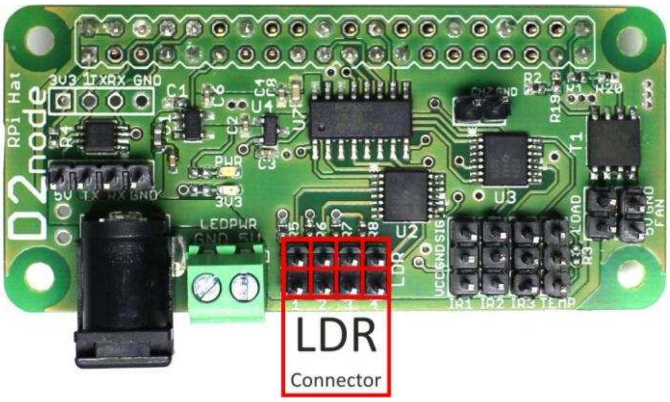
4. Electronic schematics




The Buffers for Analog Inputs For LDR
U2
MCP6044-E/ST
TLC274 will be used instead MCP604 ! ! !
The Buffers for Analog Inputs For IR distance sensors
U3
MCP6044-E/ST
The OpAmp buffers added, in case to prevent the problems with inaccurate ADC readings (due to high output impedance of sensors)
Raspberry Pi 20x2 0.1'' Connector
U5
PIZERO_THROUGH_HOLE
D2 Raspberry Pi Hat. v. 1.0. Florarobotica
D2_Rpi_Hat_v1.0
07.06.2016 10:59
Sheet: 2/2
Igor Kuksin, CYBERTRONICA RESEARCH

5. Connectors configuration

5.1 LDR Connectors

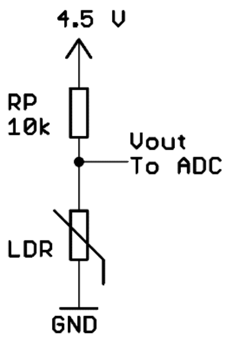


The D2 board supports the connection up to **4** LDR sensors. Each LDR should be connected to the board with individual 1x2 2.54 mm pins female header or all 4 LDRs could be connected with one 2x4 2.54 mm pins female header connector.

The connectors for each LDR marked at the board with number from **1** to **4**. And the LDR connected to ADC inputs respectively:

1 – ADC Channel **0**
2 – ADC Channel **1**
3 – ADC Channel **2**
4 – ADC Channel **3**

The each LDR have following electronic connections:



The board supposed to use with LDR with following parameters:

Resistance at the light equal to **~40-60 Ohms.**
Resistance in the dark equal to **~50-60 kOhms.**

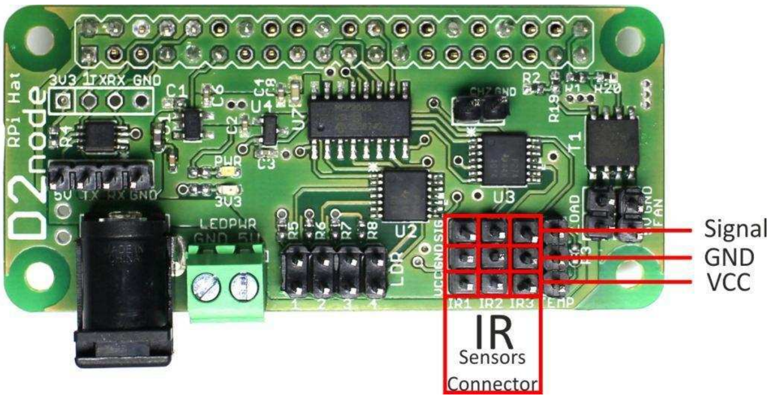
The voltage at analog input will be equal to:

Light: $Vout = \frac{R_{LDR}}{R_{LDR}+R_P} \cdot 4.5\ V = \frac{60}{60\ +10k} \cdot 4.5V =$ 26.8 mV;

Dark: $Vout = \frac{R_{LDR}}{R_{LDR}+R_P} \cdot 4.5\ V = \frac{60k}{60k\ +10k} \cdot 4.5V =$ 3.86 V;

So, the high light conditions correspond to low ADC readings, ans dark conditions to low ADC readings.

5.2 IR Proximity Sensor Connectors



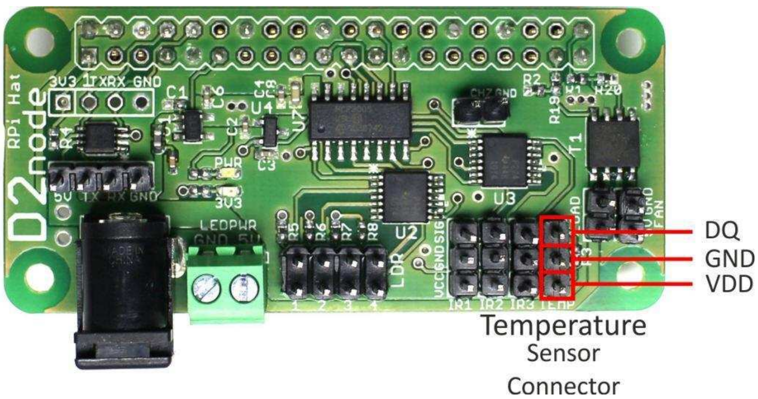
The D2 board supports the connection up to **3** IR Proximity sensors. The GP2Y0A21YK analog sensor from Sharp or similar supposed to use with this board.

Each IR sensor should be connected to the board with individual 1x3 2.54 mm pins female header.

The connectors for each IR sensor marked at the board as **IR1, IR2, IR3.** And the IR sensors connected to ADC inputs respectively:

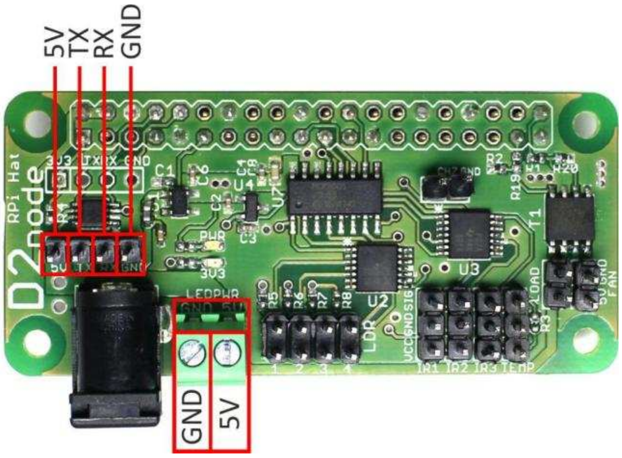
IR1 – ADC Channel **4**
IR2 – ADC Channel **5**
IR3 – ADC Channel **6**

5.3 Temperature Sensor Connector



The digital DS18B20 with 1-wire interface temperature sensor supposed to use with D2 board. The board have 4.7 kOhm pull-up resistor (according to sensor datasheet).

5.4 Pixie LEDs Connectors

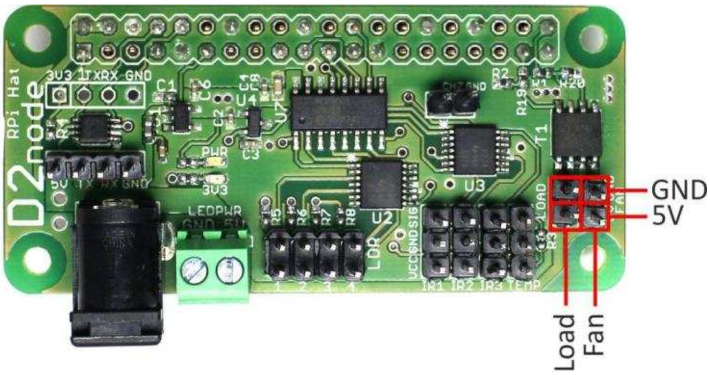


The Pixie LEDS with UART interface supposed to use with D2 board. The 3.5 mm screw terminal connector placed at the board to power the Pixie LEDs. The connector and traces at the board can handle the current up to 10 A to the LEDs. The maximum wire diameter supported by connector is 16 AWG.

Please note, to power the LEDs with such high current proper power supply to all system must be used.

The UART interface used to control the Pixie LEDs. Board have LLC converter to interface 3.3V Raspberry Pi logic to 5V Pixie LEDs logic. The have 2.54 1x4 pins header female connector used to connect Pixie LEDs UART to D2 board. 2 pins is used: one for TX line and other for RX line. Don’t use the 5V and GND pins for pixie power connection, use screw power terminal instead.

5.5 Fan and Load Connectors



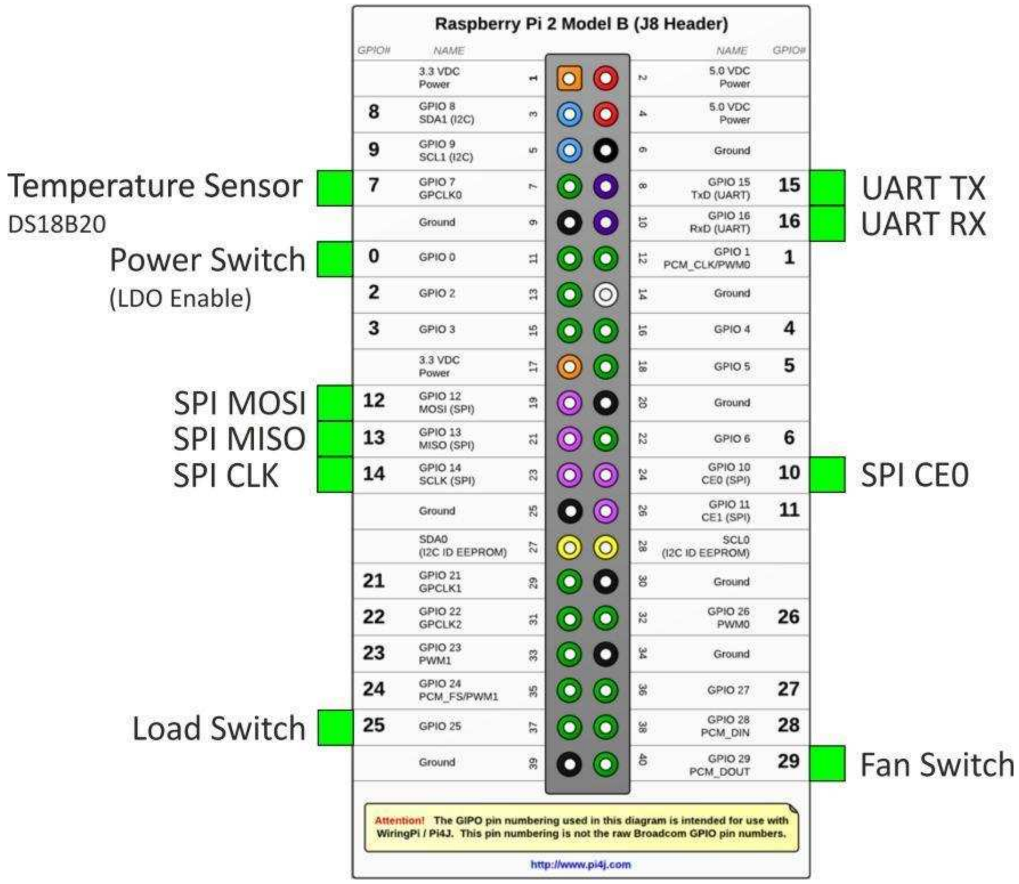
The D2 board have two channels n-MOSFET used to control the fan or other required load.
The MOSFET switch can handle a current up to 30 A. But to connector current limitation and power supply limitation the fan or load with current consumption more than 1 A **not desirable.**

Fan and load (if required) have to be connected to D2 board with individual 1x2 2.54 mm pins female header.

If required, the fan can be controlled with PWM.

6. Raspberry Pi GPIO header connector

The pins use at the Raspberry Pi GPIO header marked with green square bars with name of their function.



The Power switch used for turn on/off the power from all analog periphery at the board (ADC, ADC voltage reference, buffers, IR sensors, LDRs). The operation logic of switch is follows:

Logic **0 (0V)** – Power turned off;
Logic **1 (3.3V)** – Power turned on;

When the power is on, the blue LED at the board will indicate it.

The board have also green LED, which indicate the presence of 3.3 V (the 3.3V voltage level at D2 board is taken from Raspberry Pi). So, in the fact the green LED indicate the presence of Raspberry Pi connected to D2 board.

7. ADC

The **MCP3008** 8 channel SAR 10-bit resolution ADC chip with SPI interface used at the D2 board. The **MOSI**, **MISO**, **CLK** and **CE0** pins at the Raspberry Pi are connected to the ADC.

The **MCP5141** precise voltage reference is used for provide voltage reference to ADC. The voltage reference have **4.096 V** voltage level.

The analog inputs of the ADC are buffered with TLC274 operational amplifiers.

ADC has 10-bit resolution, so the number of ADC counts is $\mathbf{2^{10} - 1 = 1023}$. As ADC has **4.096 V** voltage reference, then ADC counts correspond to following voltage levels at analog inputs:

1023 ADC counts = **4.096 V;**
0 ADC counts = **0 V** (GND level)**;**

One count of ADC correspond to **4.096 V / 1023 = ~4 mV.**

The higher voltage level than 4.096 V will be measured as 1023 ADC counts as well.

7. Power Switch

The D2 board have 4.5 V ultralow noise LDO (Low-dropout) voltage regulator to power all sensor and analog periphery. The LP5907 chip is used, which can handle the current up to 250 mA.

The LDO also used as power switch, which can be easily turn on/off with enable pin. It used for turn on/off the power to sensors and all analog periphery at the board (ADC, ADC voltage reference, buffers, IR sensors, LDRs). The operation logic of power switch is follows:

Logic **0 (0V)** – Power turned **off**;
Logic **1 (3.3V)** – Power turned **on**;

Important note. Make sure **Power Switch** pin at the Raspberry Pi has been initialized! Don't leave it undefined. Set '**0**' or '**1**' state. This pin don't have hardware pull-down resistor. In case you will leave it in floating state, you can get unpredictable behavior. Don't do that!

When the power is on, the blue LED at the board will indicate it.

The board have also green LED, which indicate the presence of 3.3 V (the 3.3V voltage level at D2 board is taken from Raspberry Pi). So, in the fact the green LED indicate the presence of Raspberry Pi connected to D2 board.

7. List of changes