CVRA

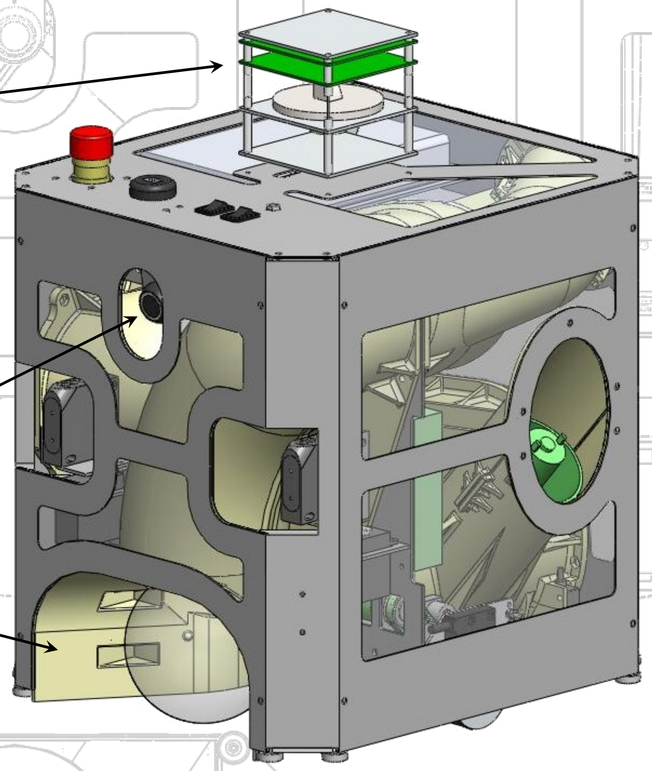
Club Vaudois de Robotique Autonome

Swiss team

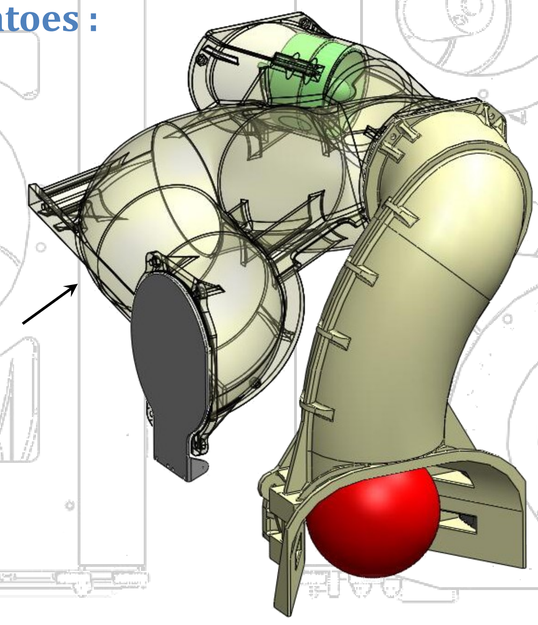
Presents :

Lollipop

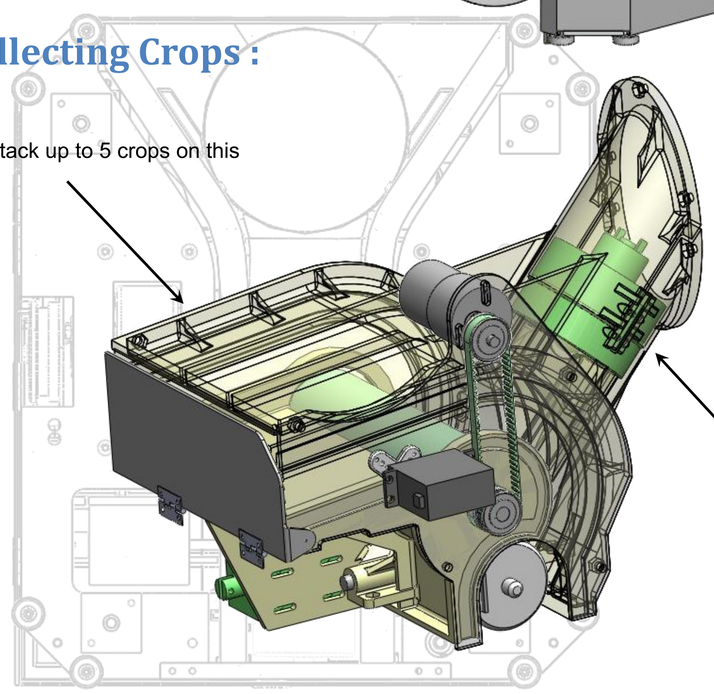
Lollipop's design:



Collecting tomatoes :



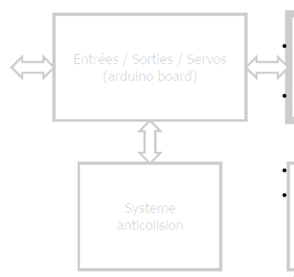
Collecting Crops :



Control architecture

To improve our didactic structure, we chose to use a control architecture which include a computer. With that, we use a polyvalent structure with different possibility like for this year:

- The motion control is on a dedicated board using a µcontroller to be able to compete on a reasonable level. We have a complete regulation over our trajectories
- Sensor and actuator are controlled by an IOs board (Arduino) so the programming task can easily be dispatched between our members
- The strategy is programmed on a small embedded computer with a Linux kernel. We use URBI, a language especially designed for autonomous robotics. The strategy is very flexible and easy to handle for more complex function
- We connect a webcam to our computer and use it to find the bad crops
- Our home made laser beacon allow us to control our position and angle on the field. We monitor the opponent position and trajectory to anticipate and avoid any collision

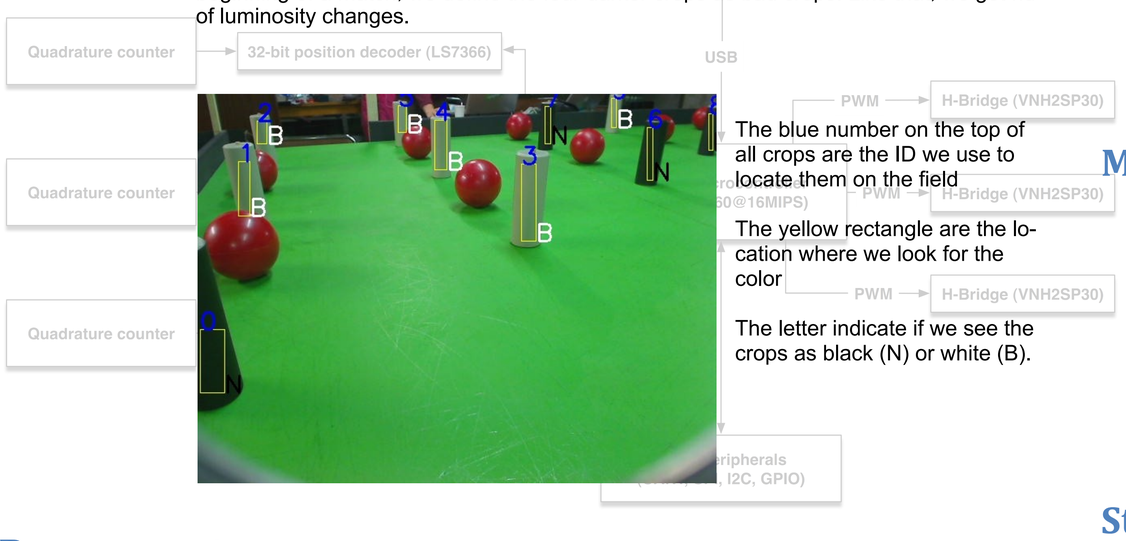


Robot pictures :

Vision system

Our vision system find the bad crop using a common webcam with a automatic gain. We use it only from specific location on the field like the start zone. Our home made algorithms, programmed on Python, are very flexible and include different functions.

We find the bad crops by sorting the luminosity of each crops found. For example, at the beginning of a match, we define the four darker crops as bad crops. Like that, we get rid of luminosity changes.



Motion control

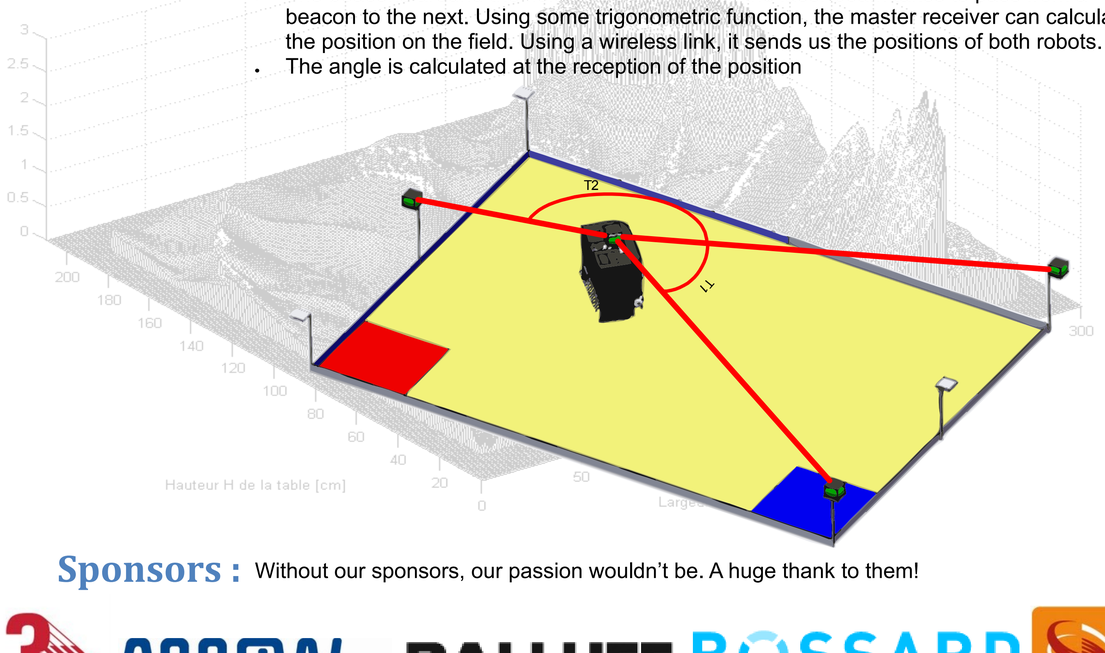
This year, we decided to design our own motion control board. This allows us to have a system that perfectly suits our needs. The essential features of the board are :

- It can handle resolver pulses up to 200 pulse per mm.
- There is plenty of connectors for more than just 2 motors (like sensors and actuators).
- The board is pretty compact
- We use an AVR microcontroller programmed in C using the Aversive framework which is a GPL development framework for robotics. Thanks to the teams who coded it !
- Quickly becomes a smoky light show if you misunderstand the polarity of the batteries (Antoine ;-))

Beacons system

The beacons system gives us the robots location on the table. The key points are:

- There are 3 receiver beacons on the side of the field linked by wire.
- Each robot has its own laser emitter beacon. On theses beacons, the laser is reflected on a cylindrical mirror to transform the circular beam to a vertical line. The mirror is mounted on a motor which rotates constantly so the laser hits each receiver.
- The beacons around the field measure the time the laser beam takes to pass from a beacon to the next. Using some trigonometric function, the master receiver can calculate the position on the field. Using a wireless link, it sends us the positions of both robots.
- The angle is calculated at the reception of the position



Strategy

This year, our robot AI (artificial intelligence) do this part of the job. It calculates the best trajectory to collect five tomatoes and five crops keeping in mind that he should arrive as close as possible to our container.
To optimize our moves, we memorize which object we picked (or tried to) and using our beacon system which object was likely collected by the other team. With these information, and our custom path planner algorithm we calculate all points of the trajectory to send them to the motion controller.
We develop our AI with URBI. This language allows us to fully simulate our robot to work on both AI and mechanics at the same time.

Members and Tasks :

Antoine Albertelli
Programmer and electronic: motion board and regulation

Cédric Debétaz
Electronic: inside process and object management

Dino Ibrahimovic
New members: helps us and learns a lot

Florian Glardon
Programmer: Strategy and URBI settings

Gil Comninellis
Mechanical engineer: mechanic prototype

Joseph Lemaître
Programmer: main computer and vision. Linux compliant.

Michael Jeanneret
Programmer and electronic: beacon system

Olivier Wenger
Programmer: interlock between our boards, communication

Romain Bersier
Mechanical engineer: mechanic designer and wiring

Vincent Kern
Motion regulation programmer
Other: poster, Media,...

All our member are needed to chose the concept of our robot

As a team, we always help each other on every task. The affectation listed are definitely not exclusive.

Sponsors :

Without our sponsors, our passion wouldn't be. A huge thank to them!