Team MegaHertz

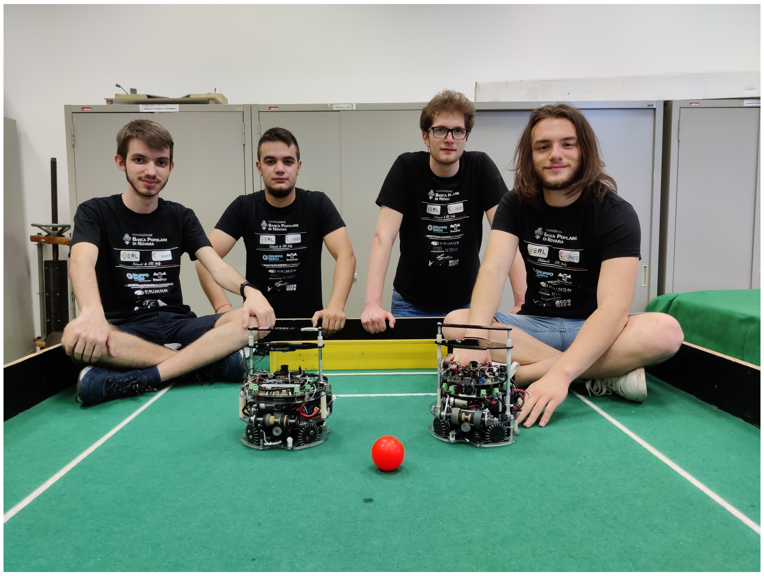
Chiorlin Davide: Software, Strategy
Schiavi Enrico: Electronics, Mechanics
Fagnani Marco: Software, Strategy
Frova Davide: Vision, Software

Mentors: Raimondo Sgrò, Alberto Mazzacarro, Alex Vellone

IIS Cobianchi and Elettra Robotics Lab
Country: Italy
Email: info@elettraroboticslab.it
Homepage: https://www.elettraroboticslab.it/
School: https://www.cobianchi.it/

Abstract: In this paper we are going to show how we developed and what are the characteristics of our new project: Viper 5. You will see how we built a robot from the ground up making it a functioning piece of hardware. You will see our new improved hardware such as the new compass (IMU), the new back dribbler, the two OLED displays for debugging purposes and the development of the two new motherboards on two layers for omnidirectional vision purposes, reducing the shadow created by the width of the robot. We implemented the fuzzy logic to create new shooting opportunities, we reduced the cycle time of the robot and tested the neuron implementation to interface with the fuzzy logic. Lastly you will see our last strategy.

1 Introduction



1.1 Team background and highlights

MegaHertz, this is our team name. It is inspired by radio waves and Guglielmo Marconi, a famous Italian scientist. Our team consists of 4 students of electronics and IT at the Lorenzo Cobianchi Institute of Verbania. Everyone is assisted by professor Raimondo Sgrò and by two associates of ERL of Domodossola, Mazzaccarro Alberto and Vellone Alex. This group has taken part in many robotic competitions, learning and discovering many things. We won the first place at the italian RoboCup junior virtual 2021 and it gave us the opportunity to take part at RoboCup worldwide virtually 2021.

2 Robot specifics

Our robots are Viper 4½ and Viper 5. Their previous version Viper 3 and Viper 4 took part at Sydney. Viper 3 and precedent version took part in Montreal, Lipsia and Nagoya World Cup, the RomeCup 2018 and Pescara RoboCup Junior 2018.
We have designed a new motherboard on those robots repositioning the components to make space for the omnidirectional vision camera. This version of the board solves the problems about the ground disturbs (Problems that we had the last year).
After last year competitions, we focused on a new version of the software which allows us to play with the omnidirectional camera also using the two football goal to orient ourselves. We did also hardware modifications to the robots with a new lower chassis, new line sensors and new roller.

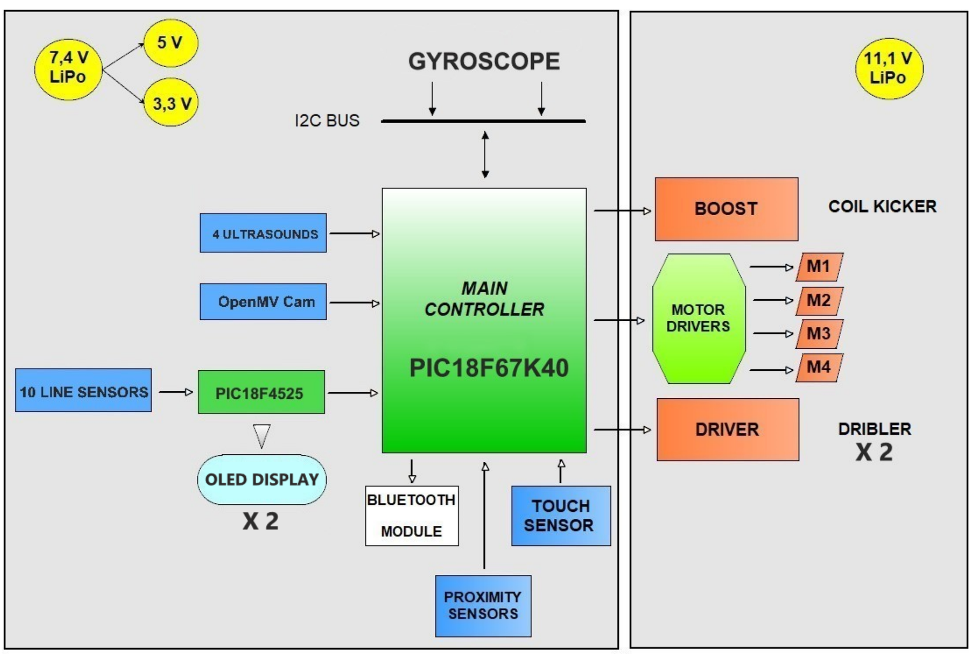


Figure 1: Viper 5 block diagram

Robots are composed by different parts, disposed on three levels.
The informations are elaborated by the motherboard, which is situated on the higher level. It is controlled by a controller PIC 18F67K40. The controller has a 64k memory and it dispose of 54 GPIO (General Purpose Input Output) pins. There are also 5 PWM (Pulse-Width Modulation) modules which control the motors, 2 Serial Ports and an I2C bus.

The last PIC18F66K80 microcontroiller was ditched because it was slower, it had less memory, it had less peripheral I/O and it had a slower 16MHz oscillator. Now the new microcontroiller allows us to use the internal oscillator up to 64MHz, now we use that at 32MHz.
Here follow the robot's controls.

- Program mode selector
- Game strategy selector
- Control and Settings button
- Indicators (LED)
- Display LCD TFT I2C x2

The motherboard manages many sensors:

- IMU
- 4 Ultrasonic sensors
- 10 White line sensors
- OpenMV Cam H7
- 3 Proximity sensor (InfraRed)
- Ball presence sensors

White line sensors and LCD Display are handled by an auxiliary controller, which is integrated on the motherboard.
There is also a Bluetooth module that allows communication between the robots or communication with a PC (Debug use only), which is very useful to view sensor's real-time value or to find hardware problems. A Serial Port handles this module. The other Serial Port communicates with the OpenMV Cam with a standard UART communication.

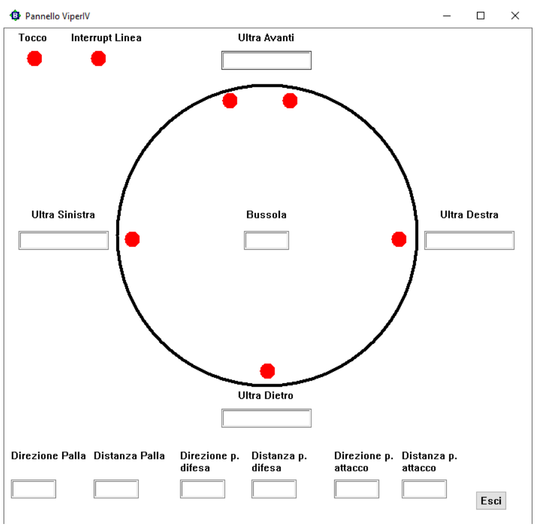
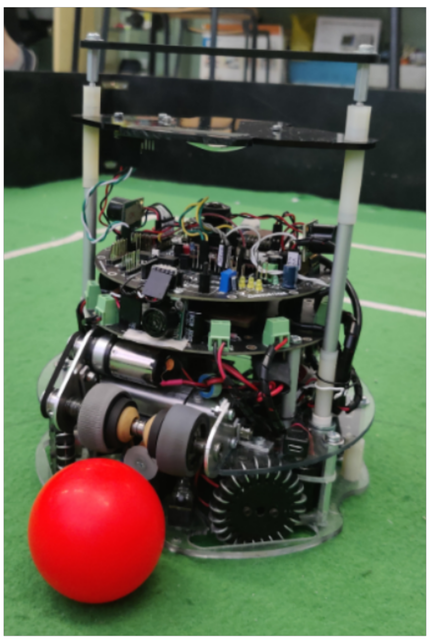


Figure 2: Debug panel

On the central level, there are situated different devices:

- Dribbler.
- Ultrasonic sensor.
- A "Boost" circuit used to kick the ball.

On the lower level, we can find motors, the "kicker" device, the white line sensor and the IR Proximity sensors.
There are two different power tensions: Logic 7,4V Li-PO (which provides 5V and 3,3V to controllers) and 11,1V Li-Po (which provides energy to motors, dribbler and kicker).
The structure was made following the RoboCup's standard dimensions (diameter, height...)



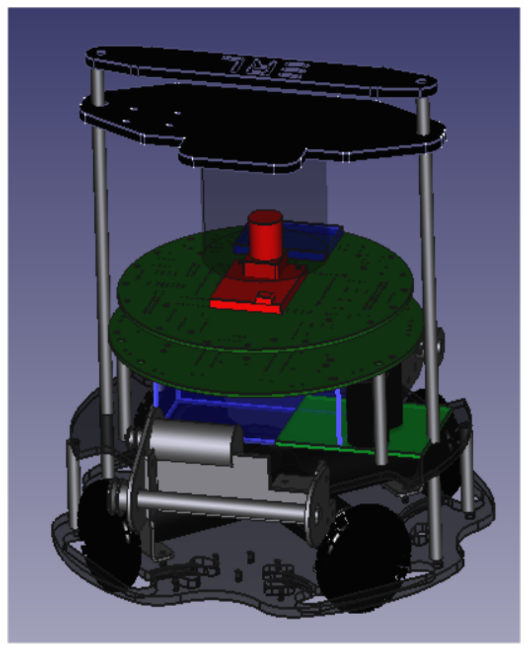


Figure 3: Viper 5 and cad diagram

2.1 Hardware

2.1.1 Back Dribbler

In Viper 5 the front dribbler is controlled using a PWM. This allows us to accellerate the ball in a controlled way to do some effected shots towards the opponent goal.
We also added a back dribbler. This allows the robot to catch the ball faster when is behind and hide the ball towards the lateral side of the field.

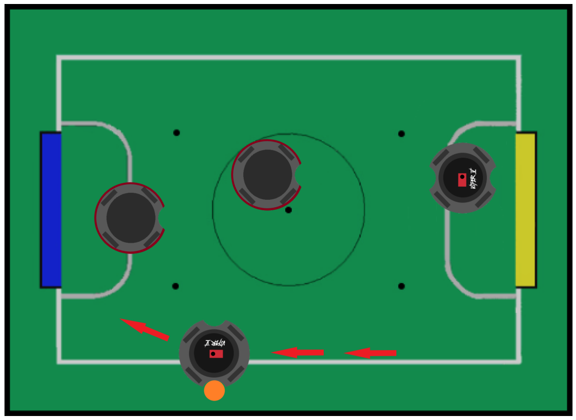


Figure 4: Diagram of the robot that is hiding the ball

2.2 Software

2.2.1 Fuzzy logic

A big problem that we faced during the competitions is that you can't predict every single condition that could happen during the match. Sometimes because of the inaccuracies of the ultrasonic sensors, like when the robot is tilted and not perfectly in line with the goals, it can't define his correct position and you can't have a fix for every single location that he could end up in.
To overcome this problem we have introduced fuzzy logic. This new logic uses continuous varialbes (from 0.0 to 1.0). With that our program can merge the data from our multiple sources and decide which is the best one to use, letting us play in a smarter way.
To use this new way of thinking we introduced a basic AI that takes 3 data: ATK & DEF goal positions, ultrasonic sensor values and the position of the robot given by the compass; then it processes all of this data and determines a float value in output between 0 and 1. This can be interpreted as a percentage of how much it is true. Then we take this value and compare it with our threshold values to determine if it's true enough to be used.
We will probably use this new way of processing the data in all the parts of code that need a not that strict decision to be made, like when we need to decide to shoot or not.

Program 1 This piece of code is implemented the classic boolean logic

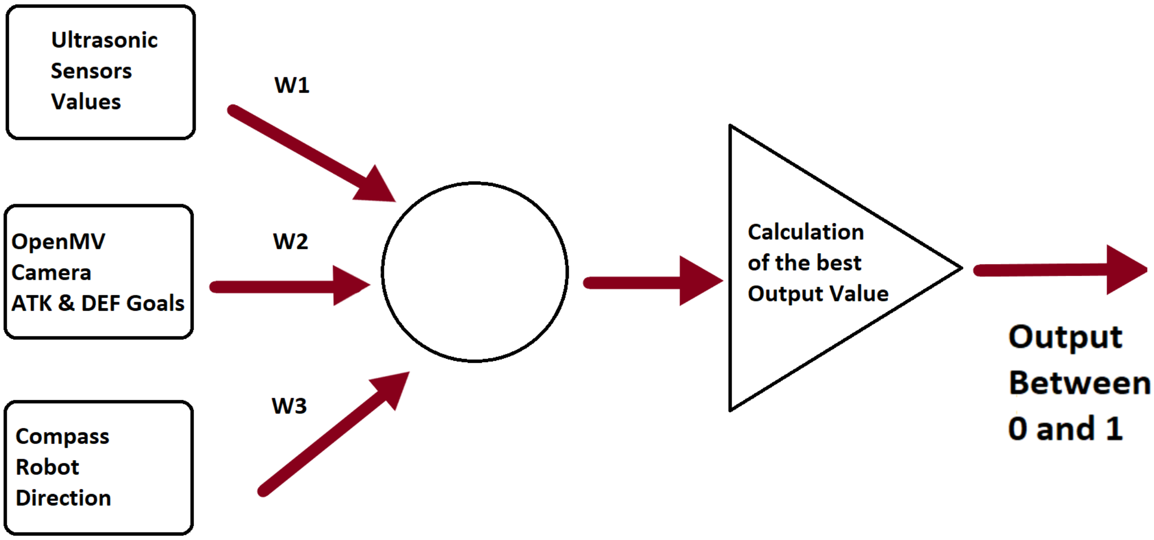
```
if ((ulDx >= SOGLIA_POS_X && ulSx >= SOGLIA_POS_X))
{
    posx=2;
}
```

Program 2 This second implementation is an example of a fuzzy implementation. The result is never false, but is always a float value from 0 to 1

```
num =(SOGLIA_POS_X-ulDx);
denom =SOGLIA_POS_X;
if (ulDx >= SOGLIA_POS_X) fuzDx=1.0;
else fuzDx = (1.0-num/denom);
```

2.2.2 Neuron implementation

This is our still in testing method of processing the data from the sensors. This is the same concept that is applied to machine learning: the inputs are from various sources that goes into a neuron and base from this data and their weights based on presets makes the robot choose between some actions.



2.2.3 Time cycle improvement

The bottle neck of the speed in our motherboard is the communication with the OpenMV Cam.
In the old protocol the time spent for the communication is 24mS for the Viper4 board. All the time is spent waiting for the response of the camera.

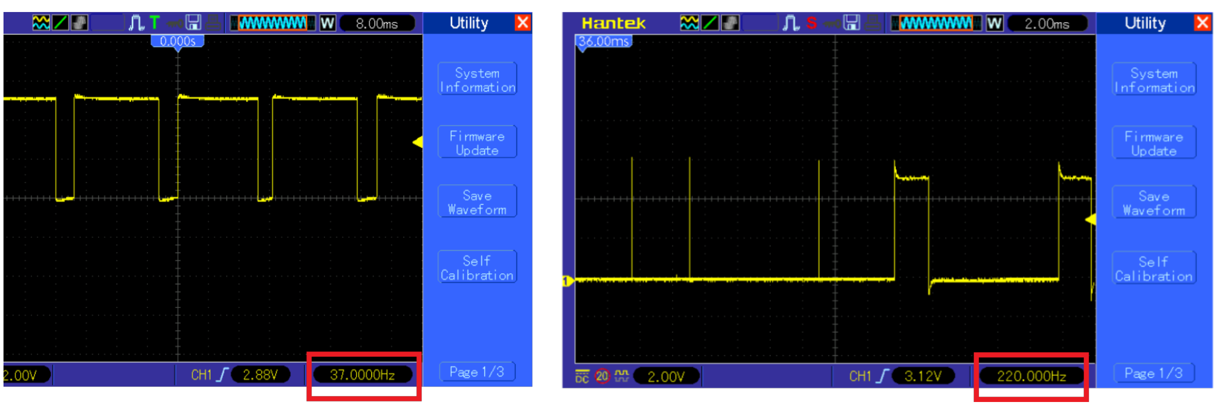


Figure 5: The high state of the DC means that Viper 4 is waiting for the camera response

In the new version of our protocol, we are not waiting anymore for the camera. We are going to receive the data only when the camera is ready to communicate. In the meanwhile, we can read the IMU sensor to better control our movements.

The rest of our main program is executed in 6ms.

2.2.4 Results

We added the back dribbler because, during our competition in Sydney we saw how much it could be an advantage on a competitive standpoint. This could also be applied when we decided to hide the ball towards the lateral side of the field.
With the new version of the serial communication protocol we improved the cycle performance from 37 cycles/s to 220. This allowed us to spend more time reading the sensors and to better handle the movements of the robot.
This new protocol summed with all the improvements of Viper 5 allowed us to get a higher score during italian soccer open: Viper 4 has scored 140 point while Viper 5 scored 158 out of 200 points.

3 Conclusions

This year, while buildinng our new robots for the competion, we have tried to unlock the potential of our robots, using different and more smart game strategies. Once we completed our strategic meeting we started improving the new robot, starting with the back dribbler and then further testing to reduce cycle time, we took our processor working frequency at 32 MHz, making the robot think faster.
We also improved our positioning system that allowes us to find where the robot is located more precisely also using the two goals recognised by the camera.

4 Future work

In the future we want to make a slimmer and lighter robot, improving speed and handling. Further more we want to improve the working frequency of the processor from 32 MHz up to 64 Mhz. We also want to finally implement the neuron to interface with the fuzzy logic that now is still in development.
Finally we want to squeeze out every bit of performance out of the two dribblers giving us an advantage during the competition.

5 References

MPLAB X Tool
Kicad downlaod
OpenMV IDE download
PIC18F67K40