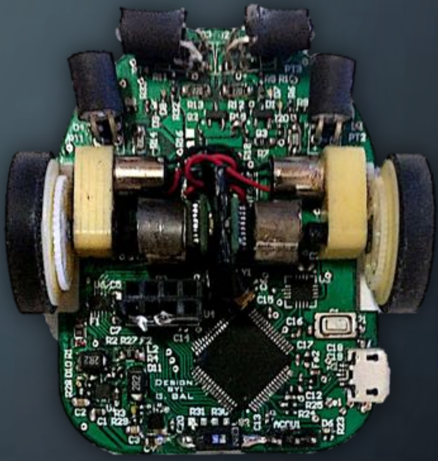
DESIGN OF AN AUTONOMOUS HALF-SIZE MICROMOUSE ROBOT

Authors:

- Balázs Göntér
- Ádám Kolozsvári
- Gergely Sántha

Supervisor: Dr. Péter Korondi



MŰEGYETEM 1782

BUDAPEST, HUNGARY

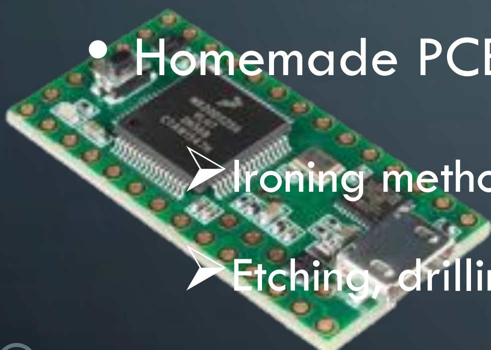
MOGI

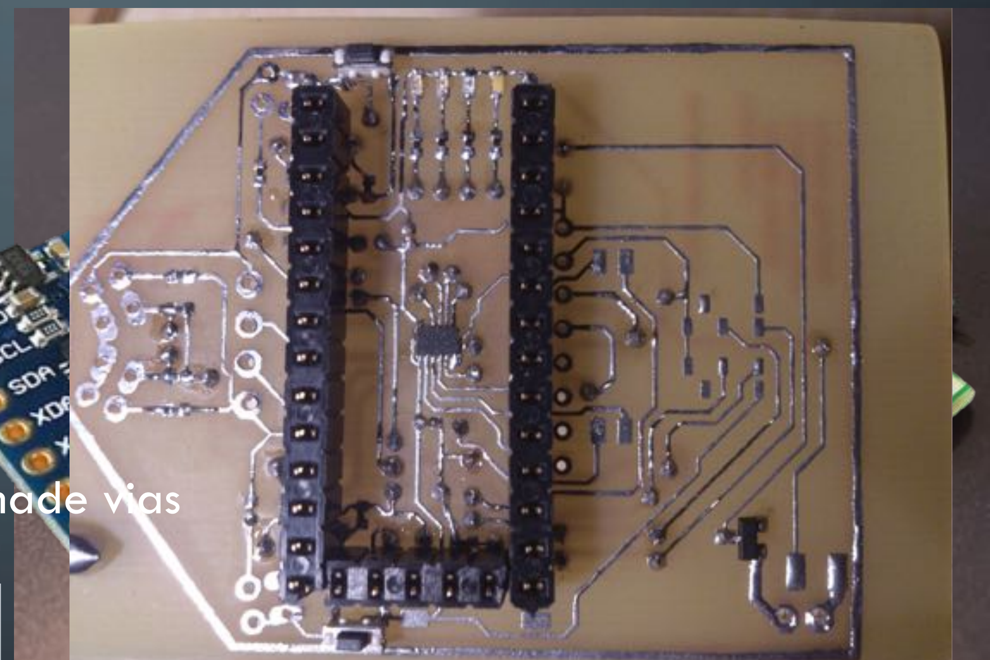
THE COMPETITION

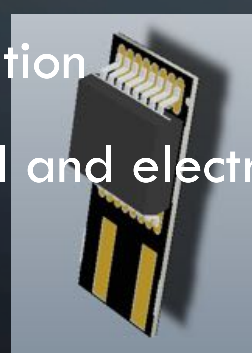
- Normalsize – 1970
- Halfsize – 2009
- Halfsize MOGI – 2015
- Complexity – mechatronics
  - ➢ Well-working powertrain
  - ➢ Reliable electronics
  - ➢ Maze solver algorithm
  - ➢ Motion control implemented on a microcontroller

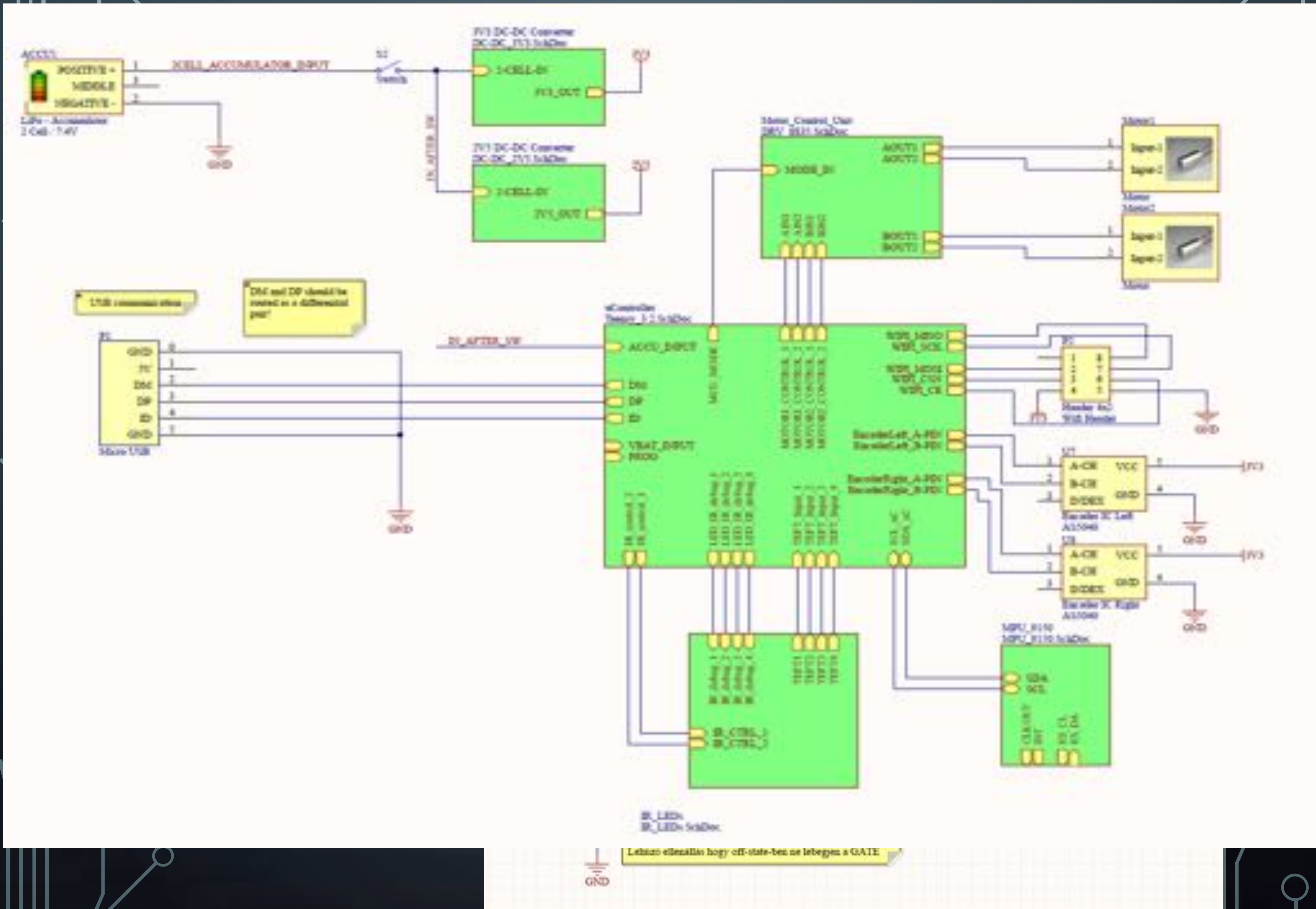
FIRST GENERATION ELECTRONICS

- Mechanics: MOGI mouse
- Modular system
- Homemade PCB
  - Ironing method
  - Etching, drilling and self-made vias
- Working regulation
- Both mechanical and electrical problems

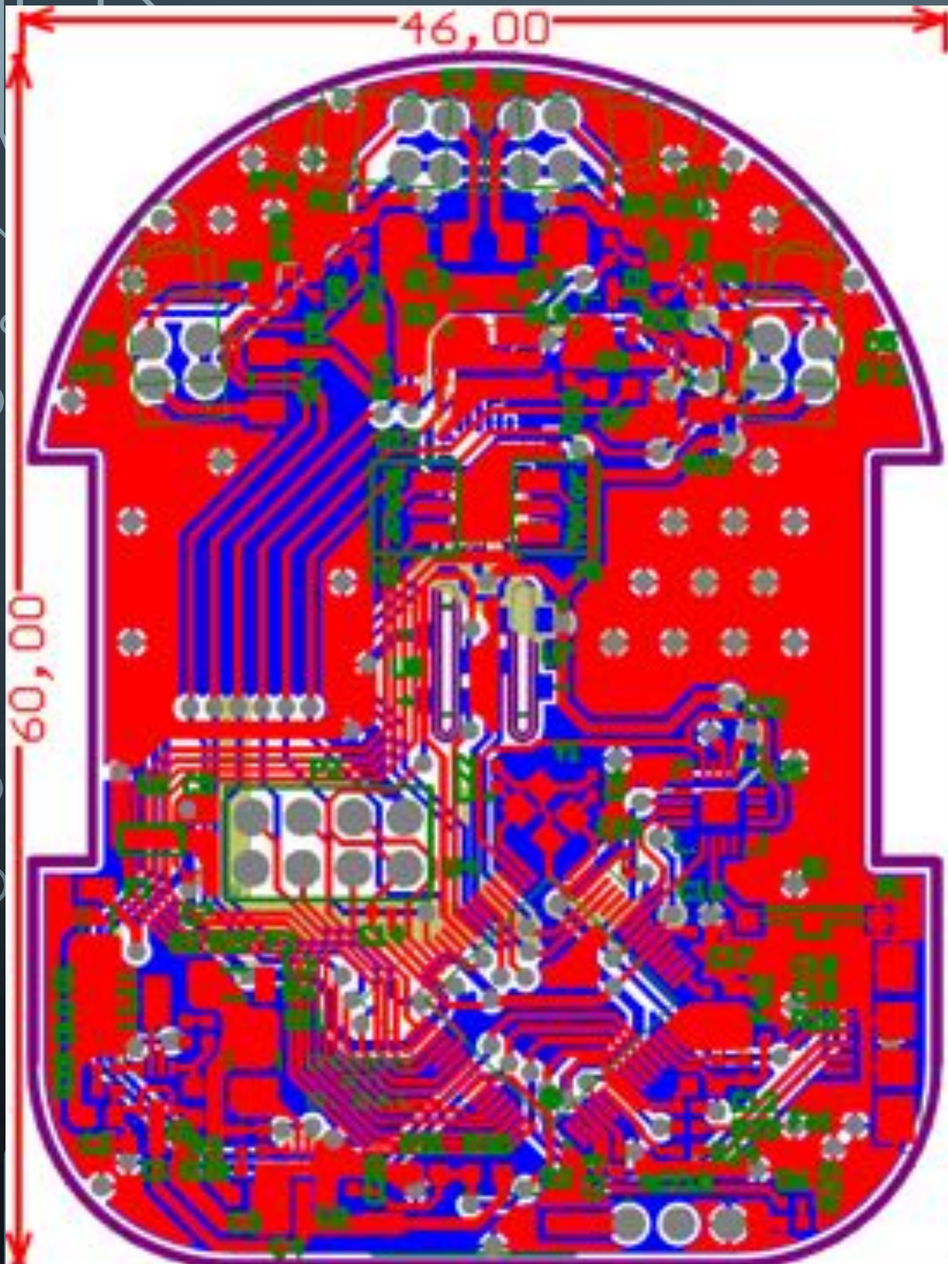


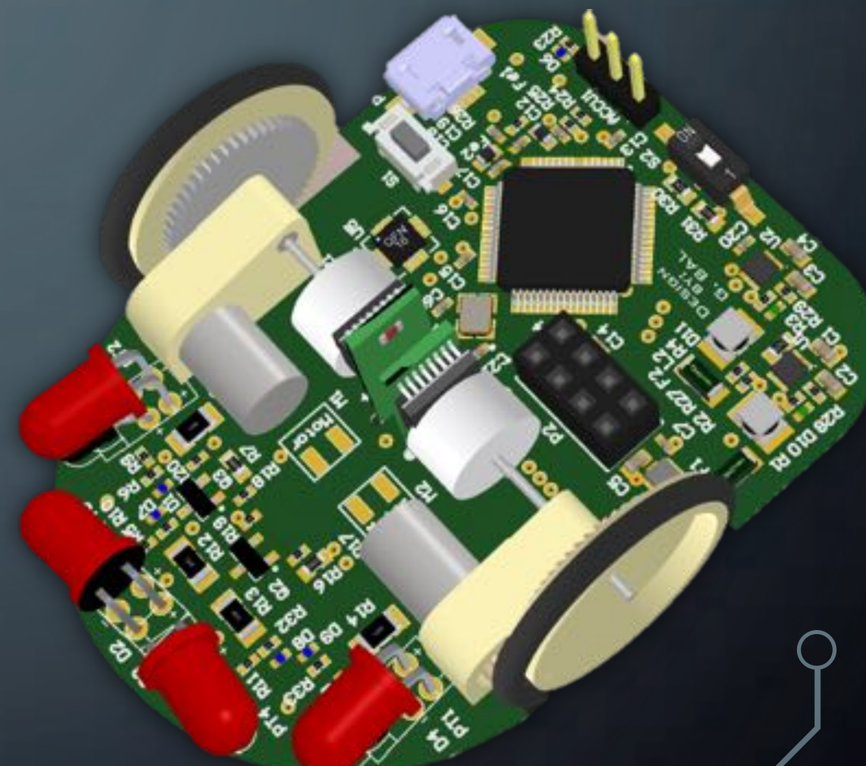






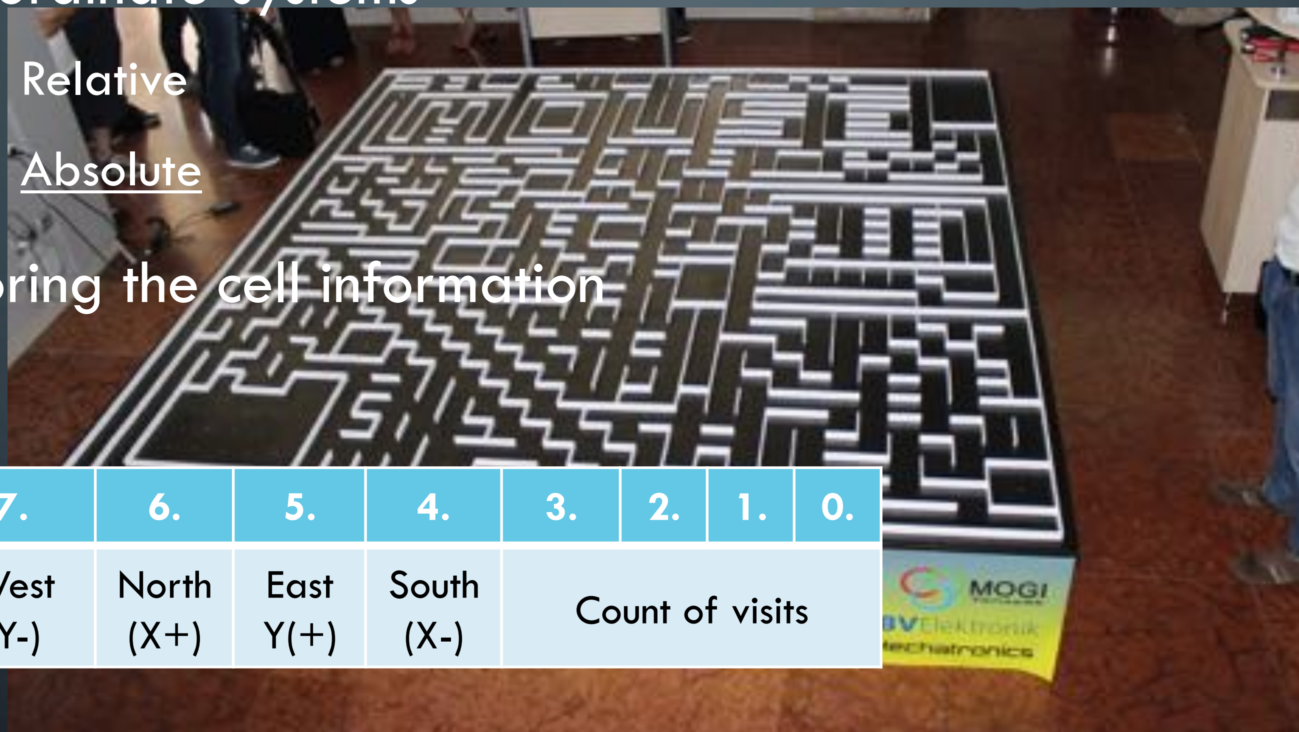
ACCU1
POSITIVE +
MIDDLE
NEGATIVE -
LiPo - Accumulator
2 Cell - 7.4V
GND
S2
Switch
Motor1
Motor2
A-CH
B-CH
VCC
GND
SDA
SCL
DM
DP
ID
GND
Micro USB
Header 4x2
Encoder IC Left
Encoder IC Right
IR_LEDs
IR_LEDs.SchDoc
Lehúzó ellenállás hogy off-state-ben ne lebegjen a GATE


46,00
60,00



RECONSTRUCTING THE MAZE

- Coordinate-systems
  - Relative
  - Absolute
- Storing the cell information

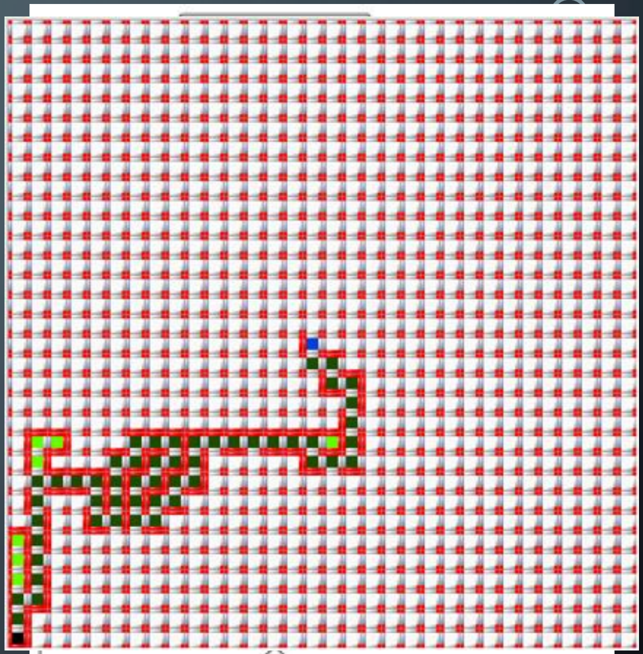


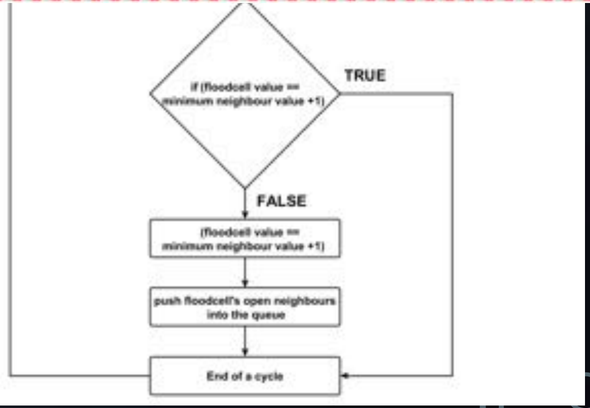
7.	6.	5.	4.	3.	2.	1.	0.
West (Y-)	North (X+)	East Y(+)	South (X-)	Count of visits			

PATHFINDING ALGORITHMS

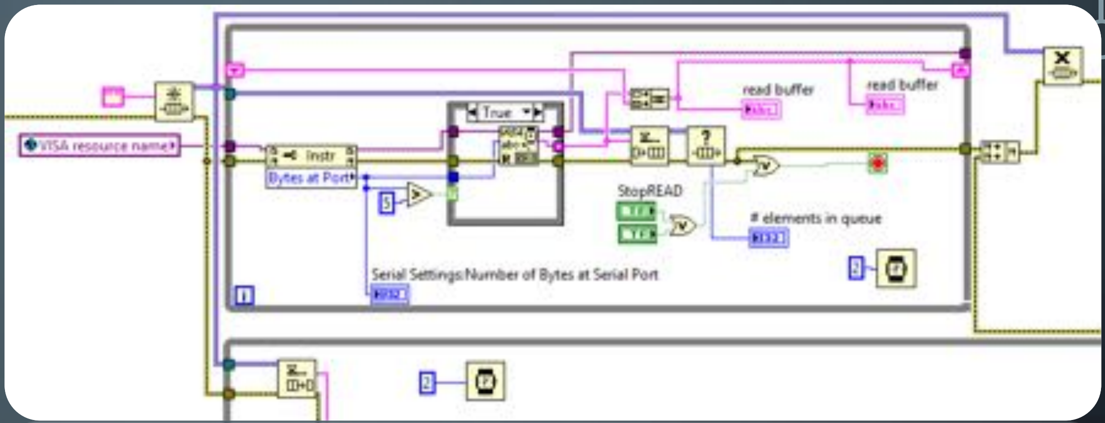
- Type of rounds
  - Search run
  - Speed run
- Algorithms
  - Without memory (random, wall follower)
  - Modified Flood fill
    - Shortest path (not certainly the fastest)
    - Low computational requirements

Required time for initialization:	3 ms
Required time to arrive to a cell	16 ms
Sum:	**19 ms**





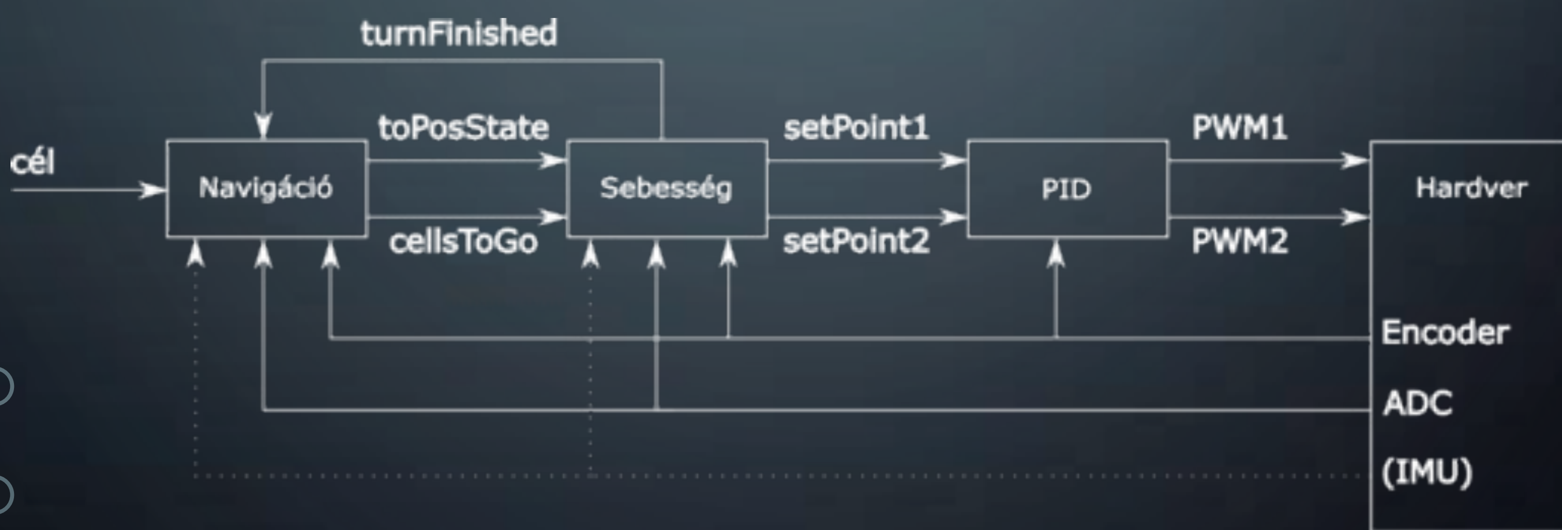
DATA PROCESSING, VISUALIZATION



- Communication
- LabView program
  - State-machine
  - Master/Slave configuration
- Data structures defined
  - Sensor data (e.g.|x511|)
  - Flood fill values (e.g.|f495@11@|)
  - Wall information (e.g.|m15@12@1001@|)
- Real-time visualization of the discovered maze
- Map generation for test purposes

IMPLEMENTATION OF CONTROL

- 3-stage cascade controller
- Timings: 6.5 ms, 3.8 ms, 1ms

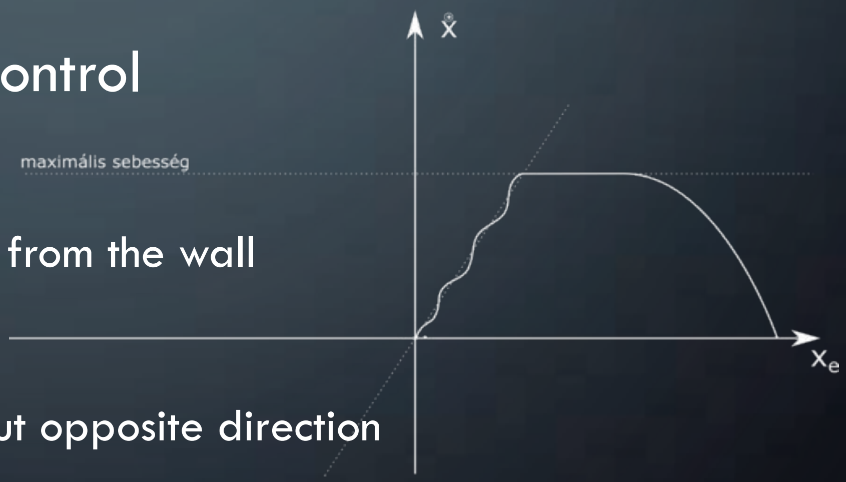


PID CONTROL

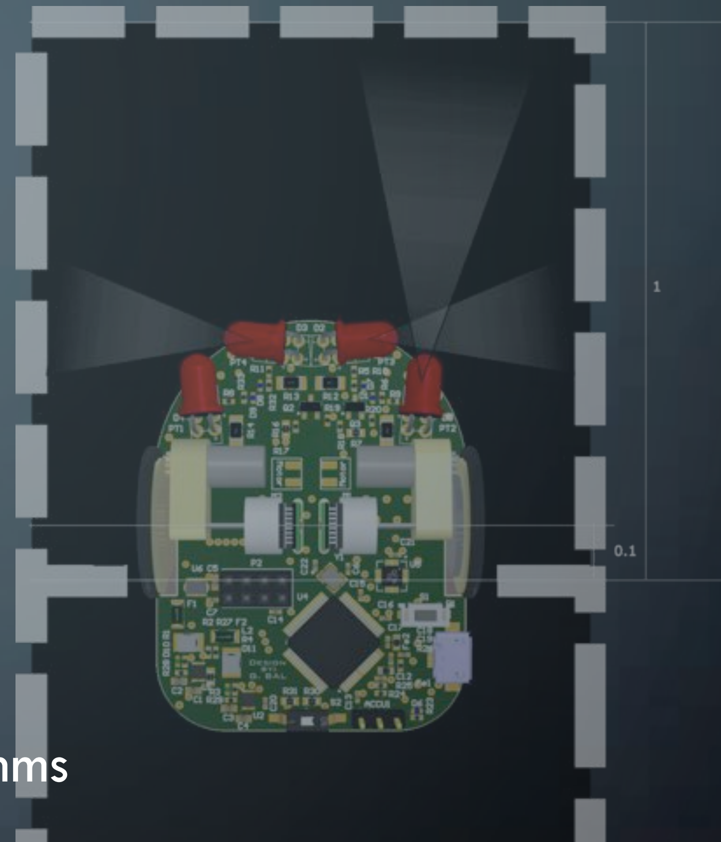
- Position control
- Parallel PD structure
- One for each side
- Gain parameters have been tuned empirically

VELOCITY CONTROL

- Not a real velocity control
- Rather it's the control of the relative position
- Simplified sliding mode control
    - Driving straight:
        - Measuring the distance from the wall
    - In-place turns:
        - Same angular speed but opposite direction



NAVIGATION



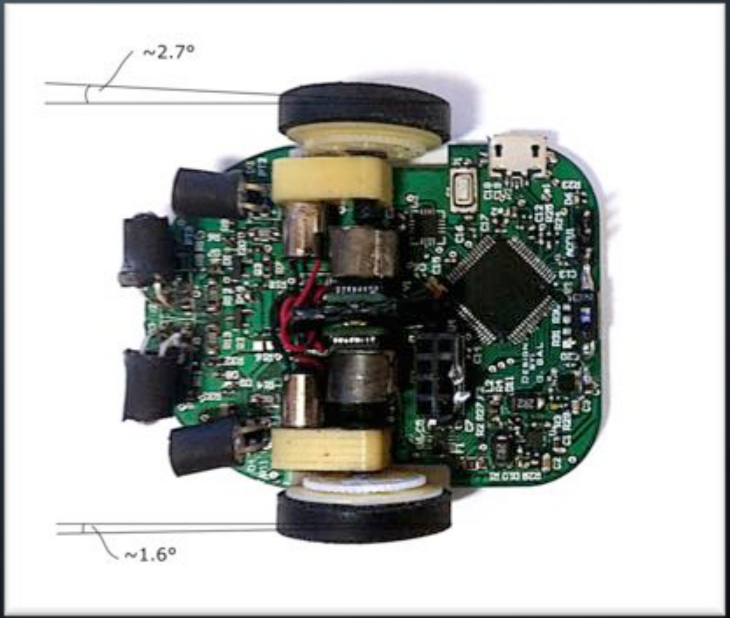
- Storing absolute position
  - Coordinates
  - Direction
- Sensing surrounding walls
- Map storing for pathfinder algorithms
- Storing wall information for adjacent cells too
- Utilizing previously known wall data
- Decision on the next step

ADAPTIVITY AND ERROR CORRECTION

- Before start:
  - Saving the initial IR sensor readings corresponding to the left and right wall distances as reference
  - Loading previously saved map (if there is any)
  - Calibrating the IMU
- During the run:
  - Keeping proper distance from the walls in case of stopping in the middle of the cell
  - Maintaining left and right distances

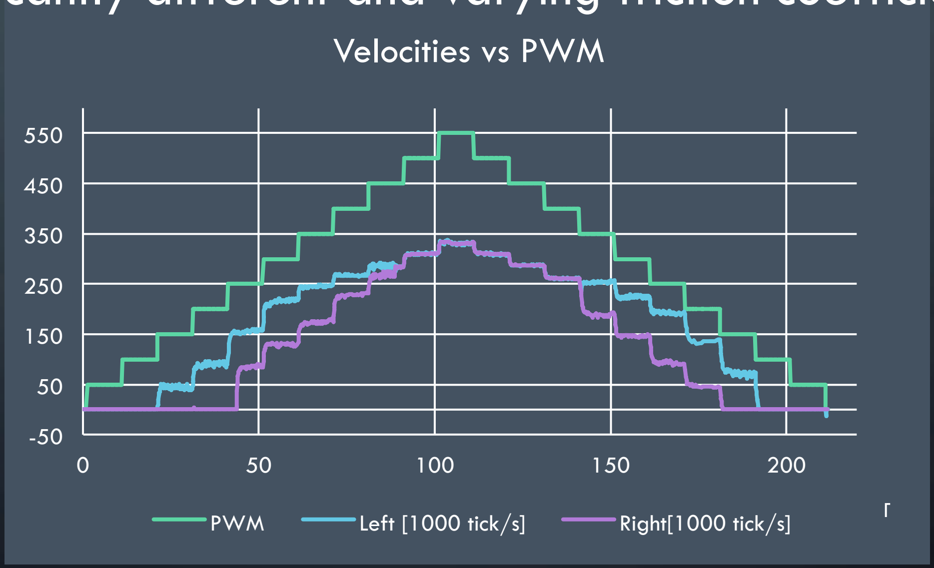
PROBLEMS WE EXPERIENCED

- Slipping of the wheels
- Inhomogeneous, varying lighting conditions
- Misalignment of the wheels from the symmetry plane



PROBLEMS WE EXPERIENCED

- Sticking drivetrain
- Speed difference between the 2 sides
  - Significantly different and varying friction coefficients



POSSIBLE FURTHER DEVELOPMENTS

- More accurate position determination
  - Fusing available sensor data with kinematical model
  - Online recalibration of the IR sensors
- Better configurability
- Dynamic turning
- Enhancing pathfinding algorithm for diagonal runs (time based)
- Hardware improvements

THANK YOU FOR YOUR ATTENTION! FOR MORE INFORMATION FEEL FREE TO WRITE US:

GONTER.BALAZS@GMAIL.COM