UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Infineon Robotic Car

Electrical & Computer Engineering

Group 30

November 30 2023

Outline:

- Objective and High Level Requirements
- Final Product and Video
- Block Diagram and Flowchart
- Design Choices
- Challenges
- Conclusion

OBJECTIVE

Create a robot to demonstrate the capabilities of Infineon's new PSoC 6 microcontroller through functions including voice command, line following and radar obstacle avoidance.

DESIGN

High Level Requirements

- **Voice Command:**
  - One word commands: "Forward", "Reverse", "Stop", "Speed up/down"
  - Control robot's motion with 90% success rate.

- **Obstacle Avoidance:**
  - Radar recognize obstacles as small as a 5cm cube
  - Turns the robotic car around
  - Range: ~ 2-10cm
  - 90% success rate

- **Line Following:**
  - Follow a 5m black taped track
  - Curvature: minimum radius of 25cm
  - Up to 70% speed
  - 90% success rate with less than 7.5cm off track

High Level Requirements - continued

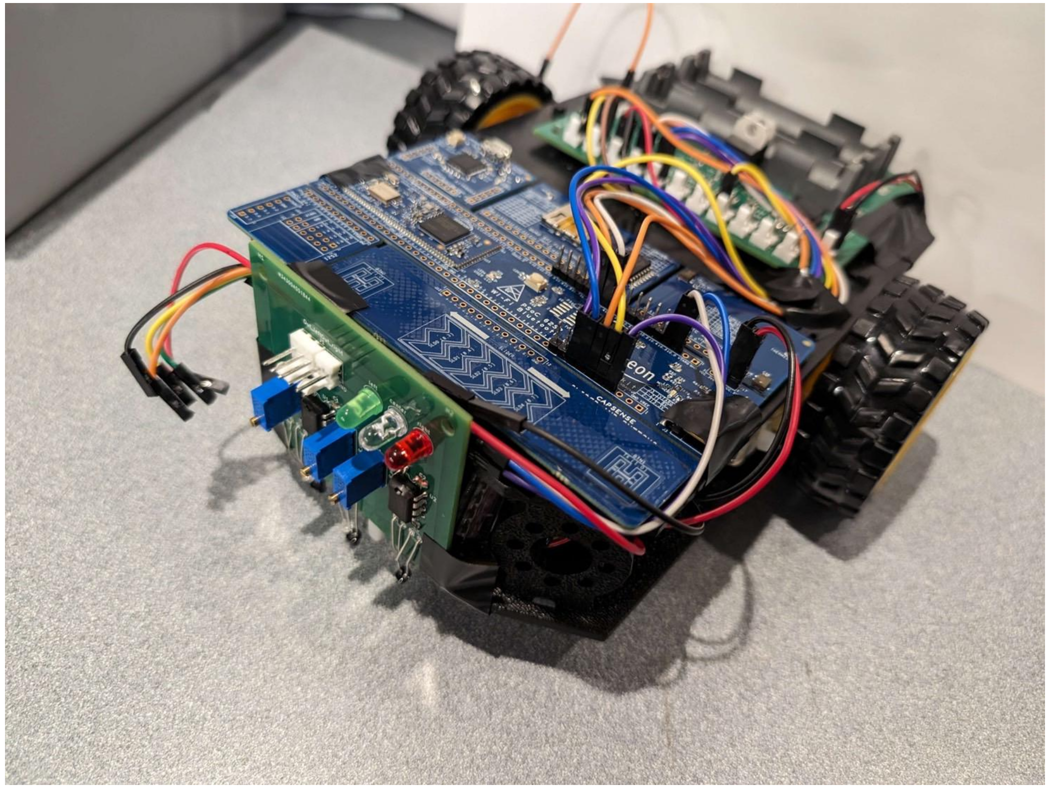
- **Speed Control:**
  - Control the motor speed continuously from 0-100%
  - Less than 2 seconds of setting time.

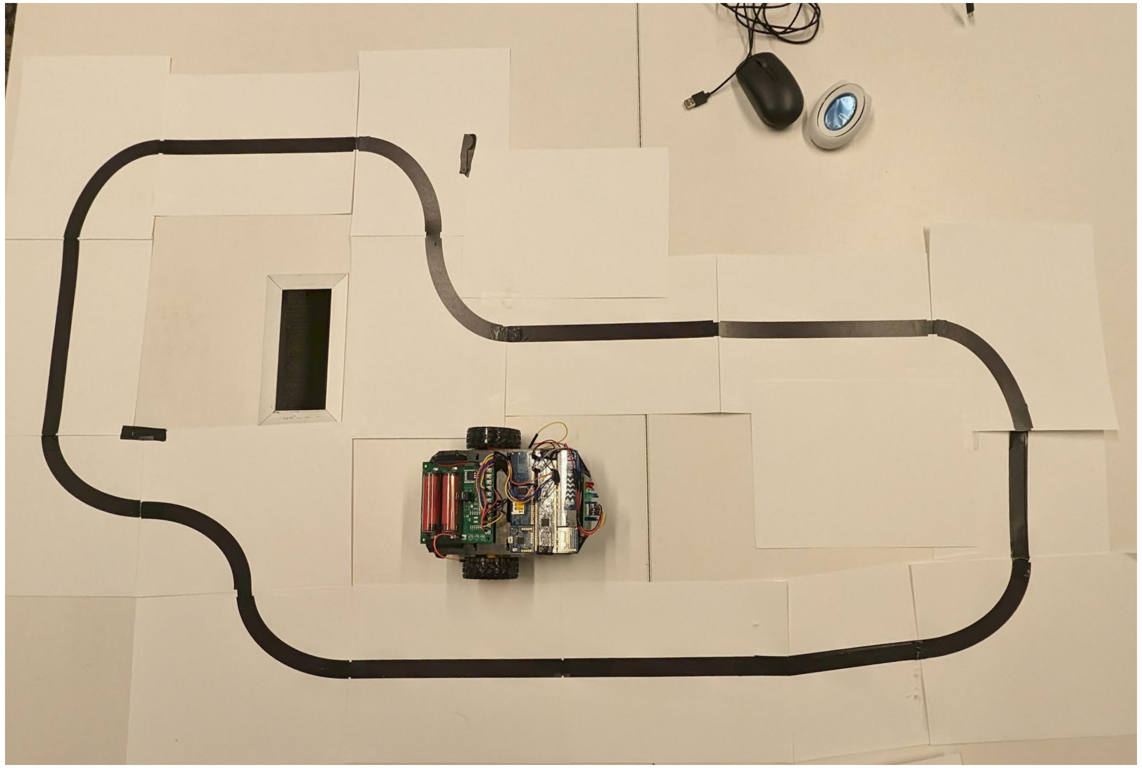
- **Battery Life:**
  - Onboard rechargeable battery
  - 90% speed for at least 10 minutes
  - Support the 15x30x5 cm car (500g)

DEMO

Here is a short clip of our project in action

Product

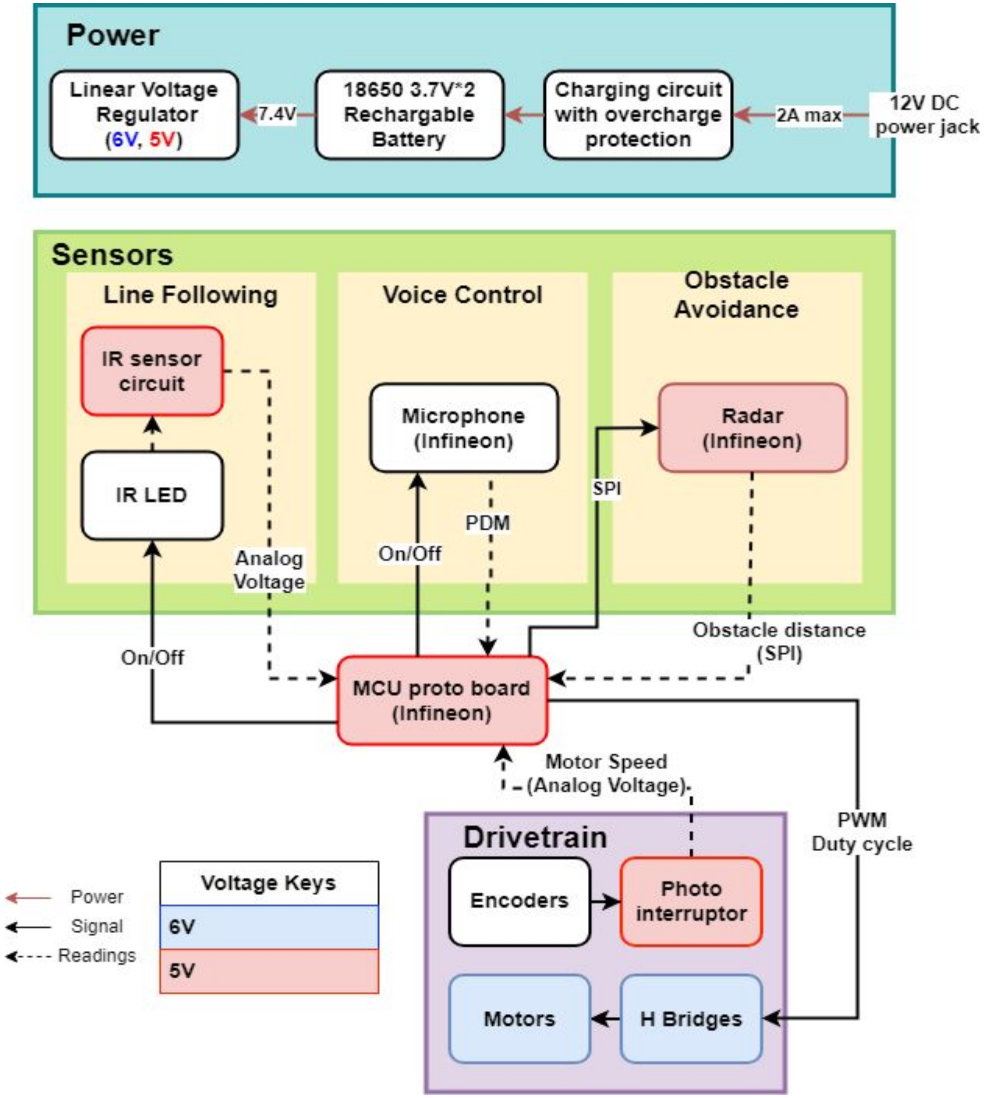




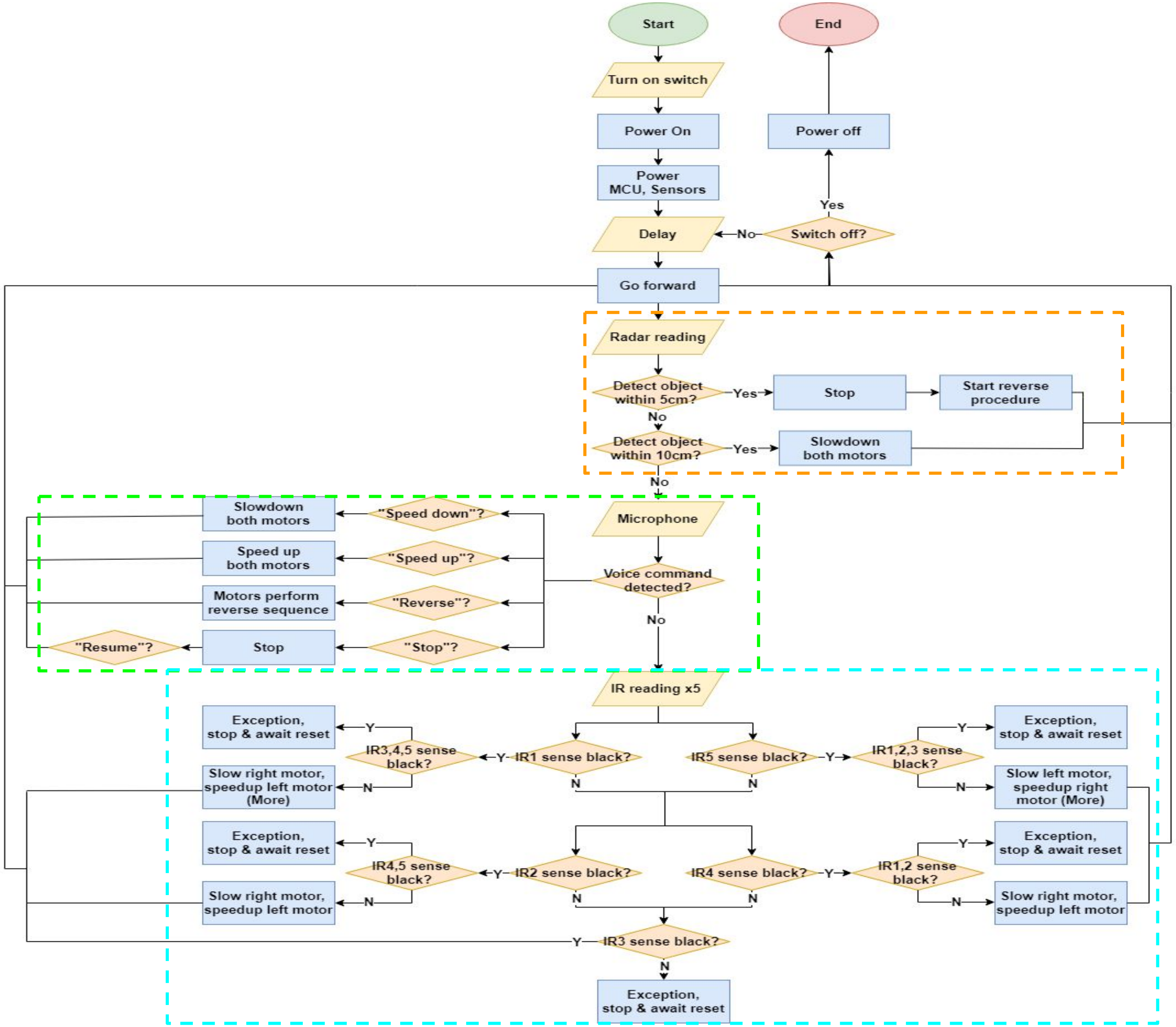
Project Demo

Robotic car following line, responding to five different voice commands, and avoiding obstacles

Block Diagram



Control Logic:

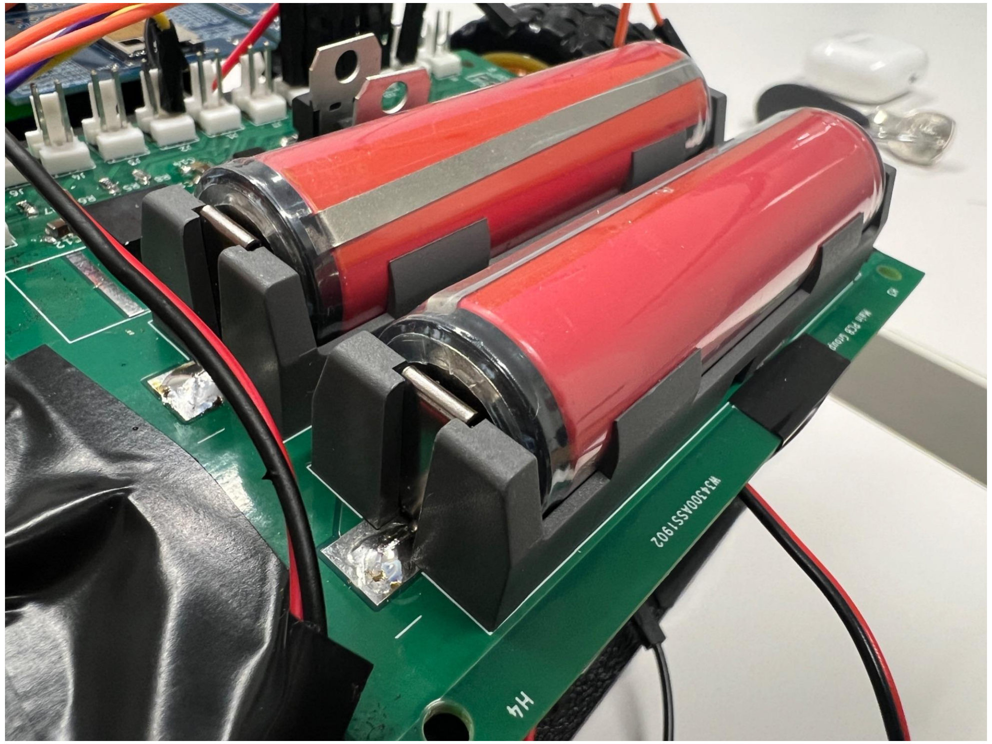


SUBSYSTEMS

Modifications from Design:

- **Power:**
  - Removal of Recharge Circuit due to safety concerns with cell balancing
- **Microcontroller:**
  - Removal of Reverse capability due to inconsistent line following
- **Sensor:**
  - Lowered number of IR sensors from five to three since there was no benefit brought by added sensors
- **Drive Train:**
  - Removal of closed loop speed feedback control due to low motor speed

Power Subsystem



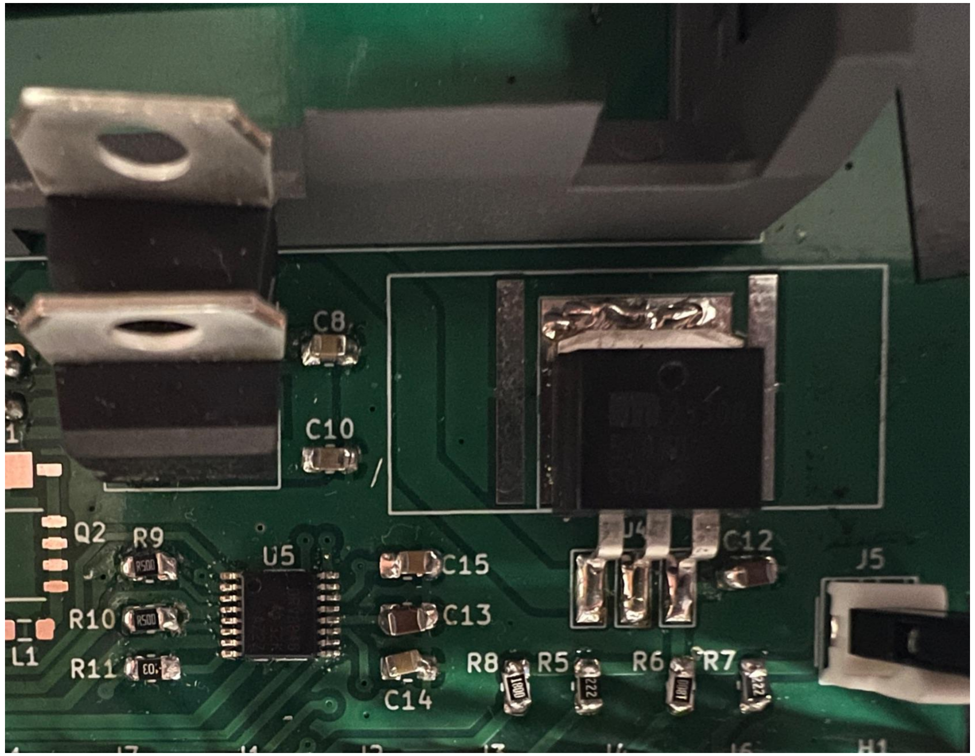
Rechargeable batteries as power source

- Two 3.7V lithium ion rechargeable batteries connected in series with combined output of 7.4V
- Has a much larger life time than initially anticipated
- Surface mounted battery holders

Voltage Regulators

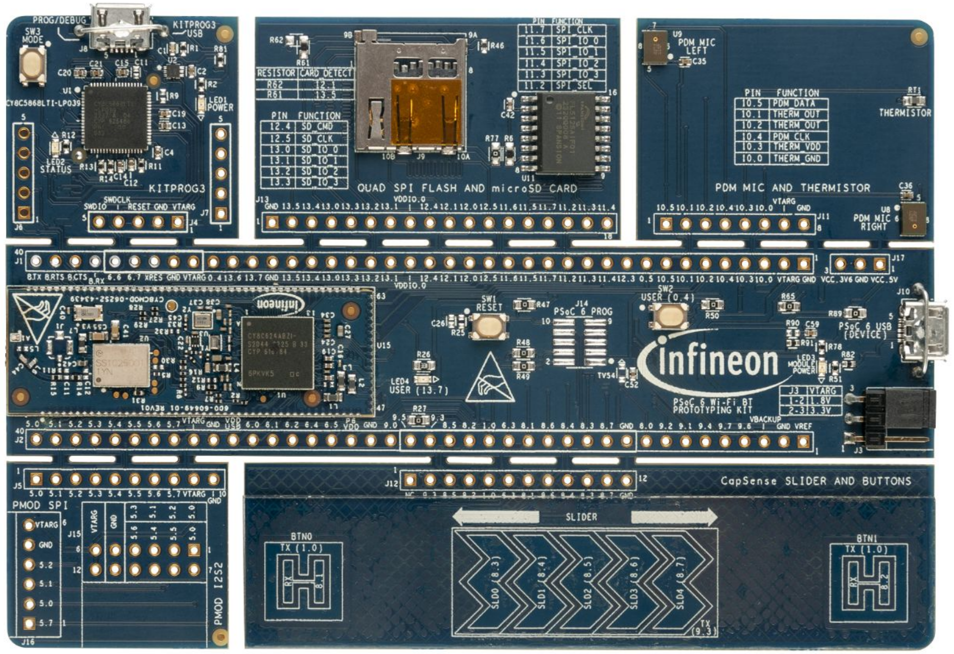
Provide correct voltage to different chips

- Convert 7.4V to 6V and 5V
- Two 6V 1A regulators in parallel for motors
- Single 5V 3A regulator for all other parts of the circuit



Microcontroller Subsystem

Microcontroller Dev Board

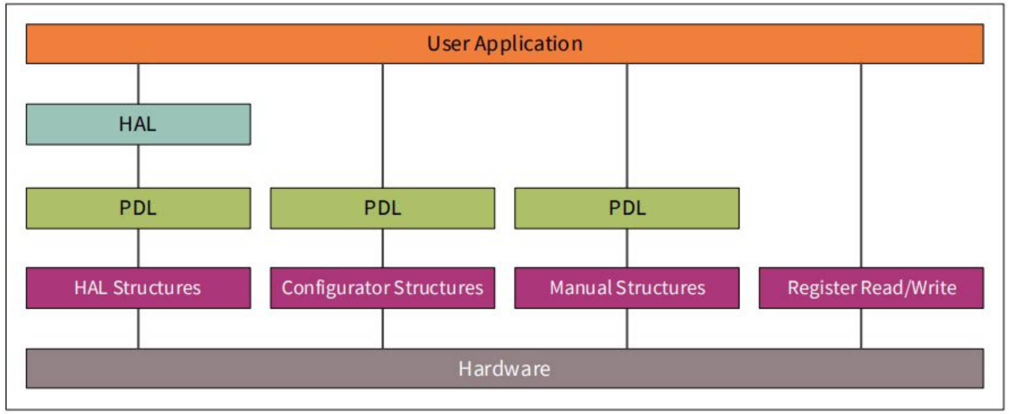


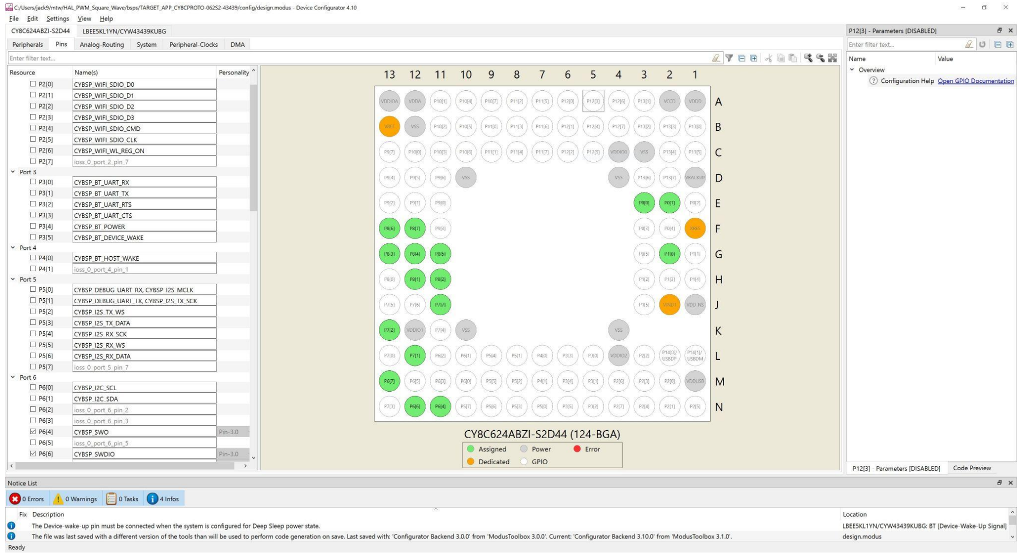
The microcontroller should be able to take readings from all sensors and determine how to change the motor behavior

- Output from sensors:
  - radar - SPI
  - microphone - PDM
  - Analog voltage - IR sensors

- Modularized programs for each function:
  - Voice command and radar might pause the other modules → Threading

Programming

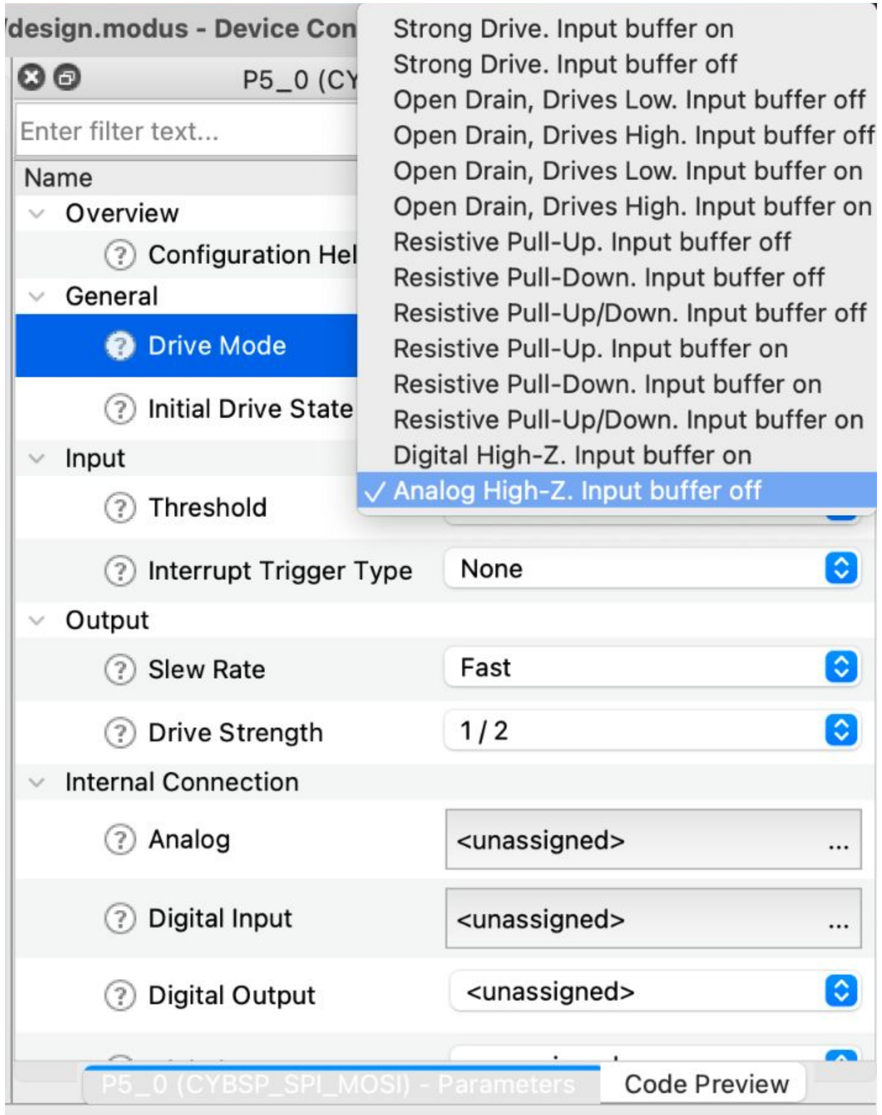




Modus Toolbox's Eclipse IDE from Infineon

- HAL(Hardware Abstraction Level) vs PDL(Peripheral Device Library)
  - HAL: generic, portable among different product families
  - PDL: better control over individual registers
- Device configurator: generate code (header) from pin definition
  - Enabling resource → conflict with HAL

Programming Contd.

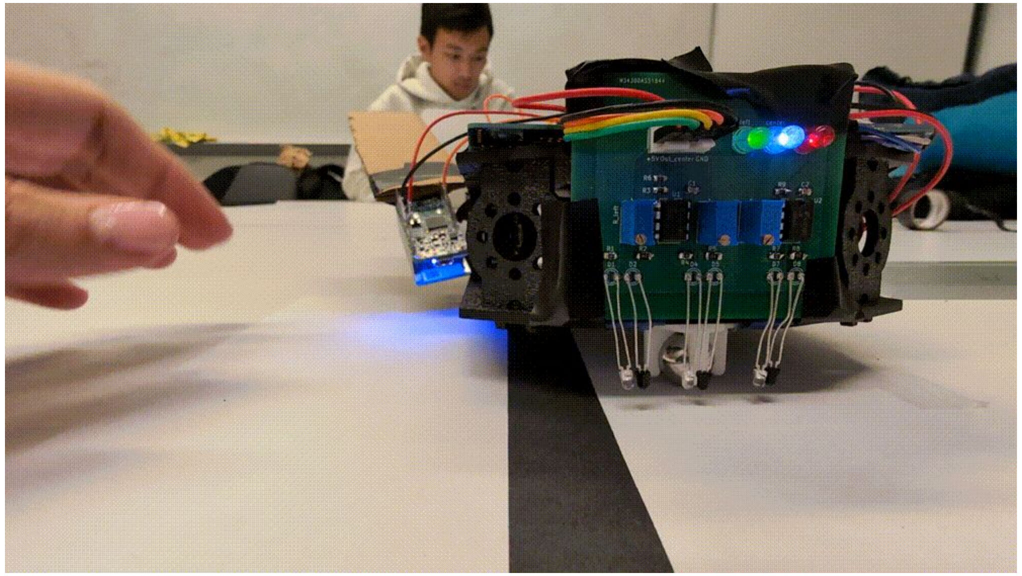


Drive Modes and IR Logic

- Output from IR sensors are analog
  - Used Pull Down Drive mode to convert to digital
- Output from microphones are PDM signal
  - Used Digital High-Z drive mode to preserve digital signal
- Output from radar is digital
  - Used Digital High-Z drive mode to preserve digital signal
- Output from Photointerrupter is analog
  - Used Pull Down Drive mode to convert to digital
  - Interrupts program to handle RPM
- infinite for loop
  - Updates current IR states and voice command
  - Four if statements for four speed states
  - Base duty cycle speed of 50% updated by voice commands

Sensor Subsystem

IR sensors



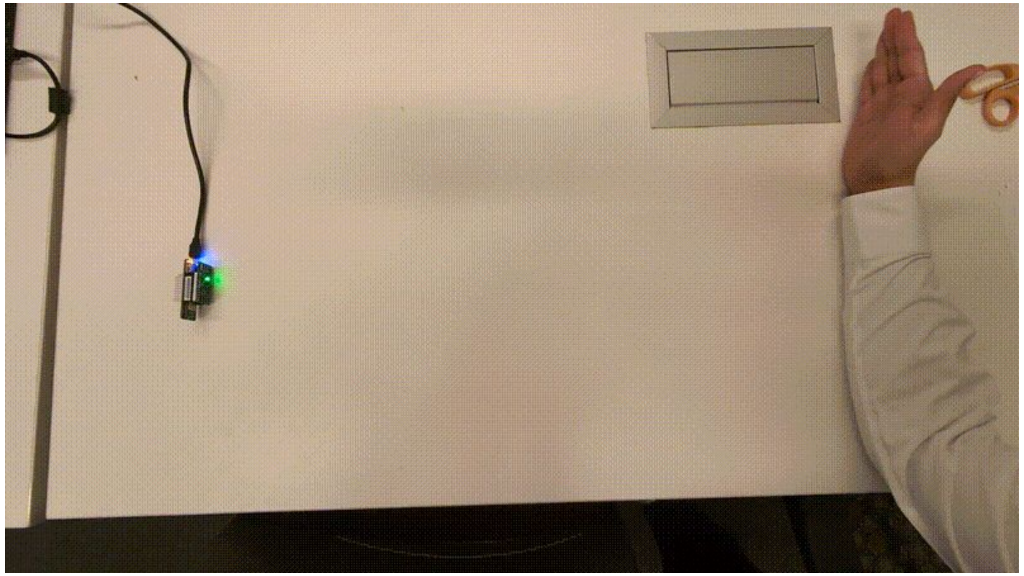
IR sensors should be able to tell the MCU the relative location of the car to the track

- 5 pairs of IR LED (emitter) and photodiode (receiver) spaced apart wider than the track.
    - 3 pairs were enough for line following
- Outputs analog high (~3.3V) when the pair is above the white background and low when above the black track
    - Added LEDs to indicate output of each sensor pair
- Tunable range
    - Trimpot as voltage divider

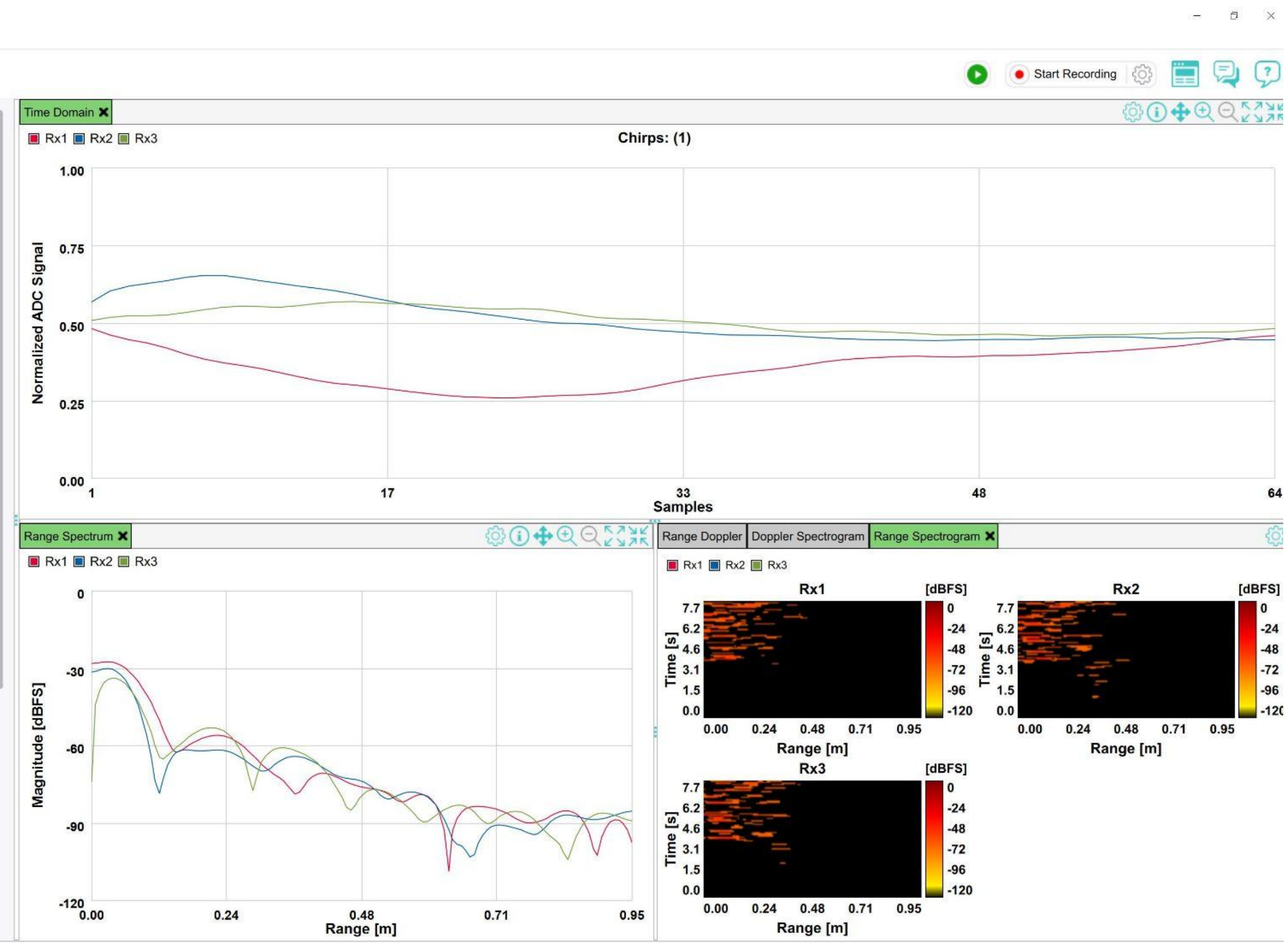
Detect obstacles in the way of the car

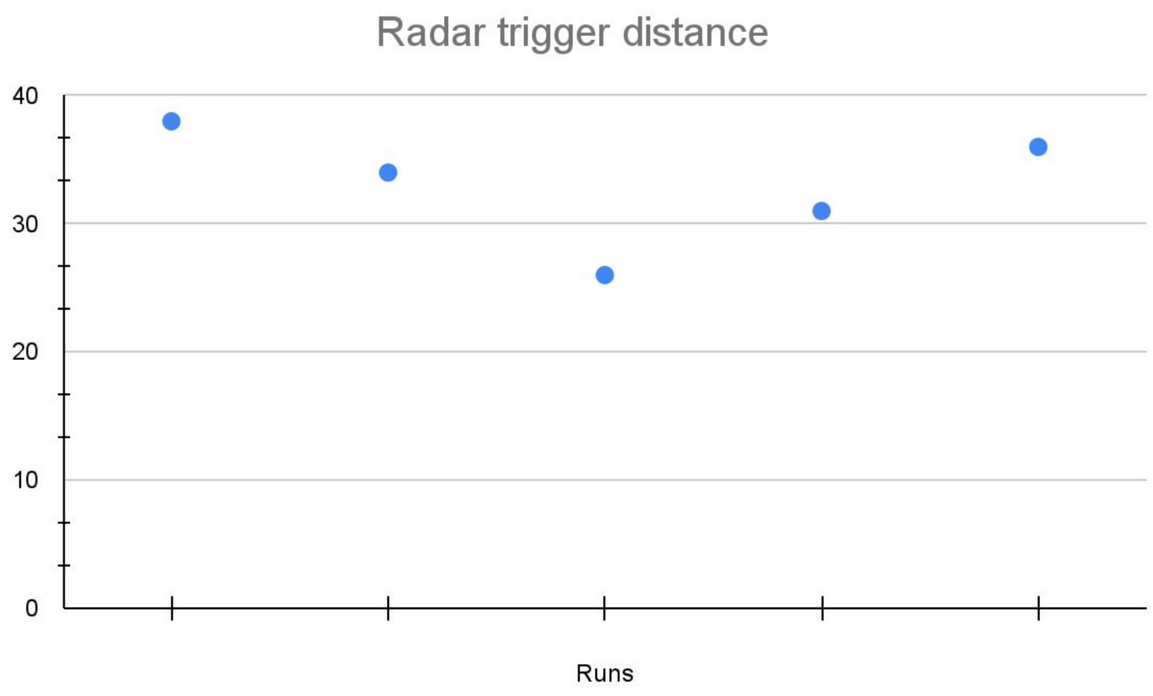
- Detect size: 5cm block
- Detect range: 10cm
- Car to turn around and follow the track in the opposite direction when an obstacle is sensed

```
static const xensiv_radar_presence_config_t default_config =
{
    .bandwidth                         = 460E6,
    .num_samples_per_chirp             = XENSIV_BGT60TRXX_CONF_NUM_SAMPLES_PER_CHIRP,
    .micro_fft_decimation_enabled      = false,
    .micro_fft_size                    = 128,
    .macro_threshold                   = 2.0f,
    .micro_threshold                   = 12.5f,
    .min_range_bin                     = 1,
    .max_range_bin                     = 1,
    .macro_compare_interval_ms         = 250,
    .macro_movement_validity_ms        = 1000,
    .micro_movement_validity_ms        = 4000,
    .macro_movement_confirmations      = 0,
    .macro_trigger_range               = 1,
    .mode                              = XENSIV_RADAR_PRESENCE_MODE_MACRO_ONLY,
    .macro_fft_bandpass_filter_enabled = true,
    .micro_movement_compare_idx       = 5
};
```

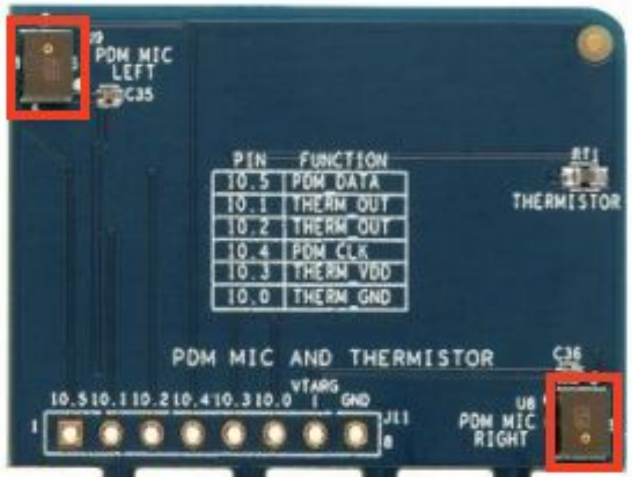


Radar





AVG: 33cm



Machine Learning Solution for Voice Control

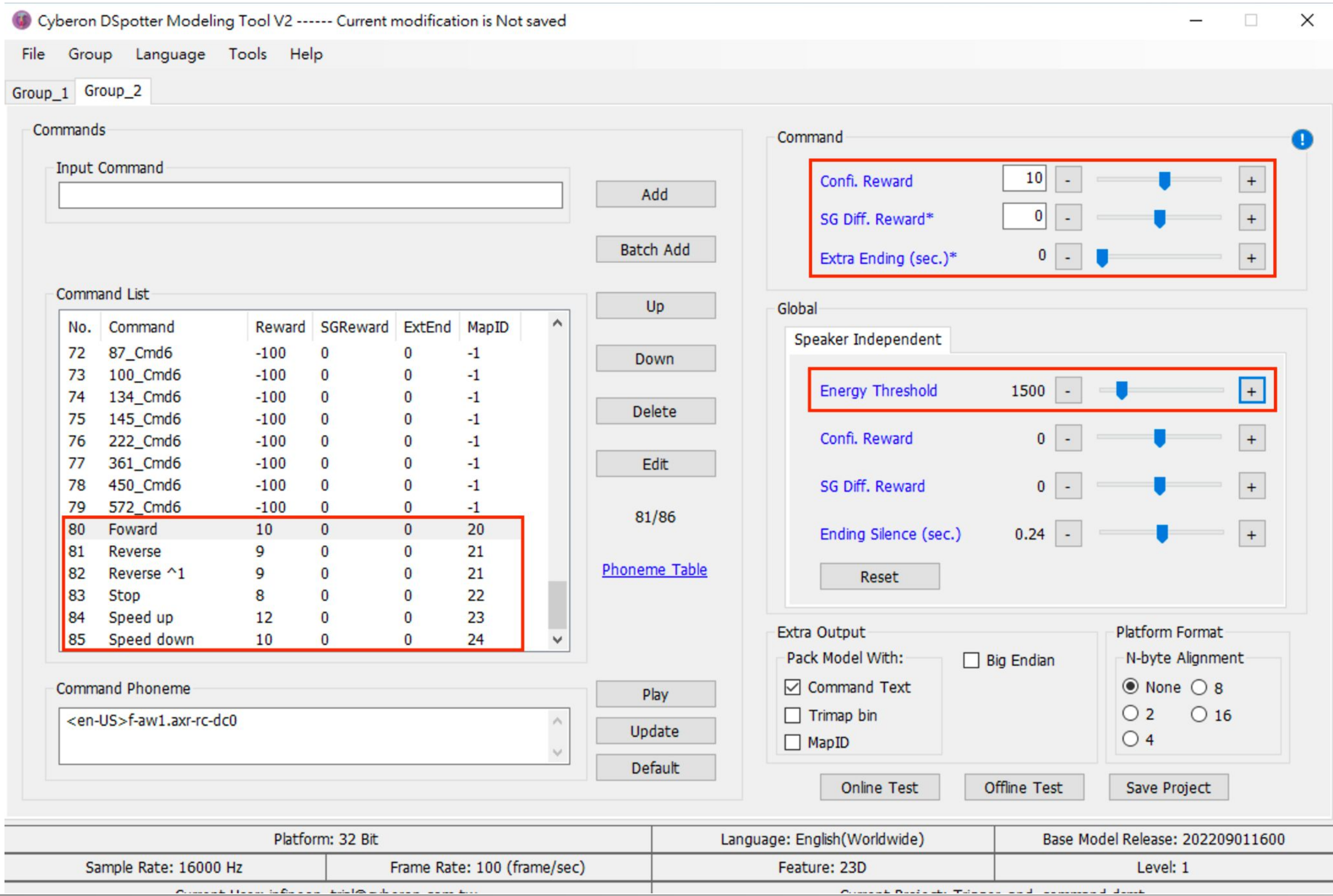
- **Two PDM Microphones**
  - Detect voice signal 2m away
  - Detect voice commands with motor noise

- **On Chip Machine Learning Algorithm**
  - Cyberon Dspotter Voice Trigger Solution
  - Detect the 5 different voice command with 95% accuracy alone

Voice Control - Software

Cyberon Dspotter: Customize Model

- **Two-stage keyword detection**
  - Trigger
  - Action Commands
- **Detection Algorithm**
  - Divide word into phonemes
- **Parameters**
  - Confidence Reward
  - SG Difference Reward
  - Energy Threshold
  - Ending Silence
  - Extra Ending



Cyberon Dspotter: Integration

- **Libraries**
  - cyberon_asr.h
  - CybModelInfo.h
- **API**
  - cyberon_asr_init(asr_callback)
  - cyberon_asr_process(pdm_pcm_buffer, FRAME_SIZE)
  - pdm_pcm_flag
- **Command ID**
  - Determine the command for action control

Drivetrain Subsystem

Provide bi-directional capabilities to the DC Motors

- Single H-bridge IC is capable of controlling two motors saving space on PCB
- Four different modes of operation, we use forward/backward PWM, fast decay mode to enable quick responsiveness of the motors
- Motor speed and direction controlled by PWM signal provided by microcontroller

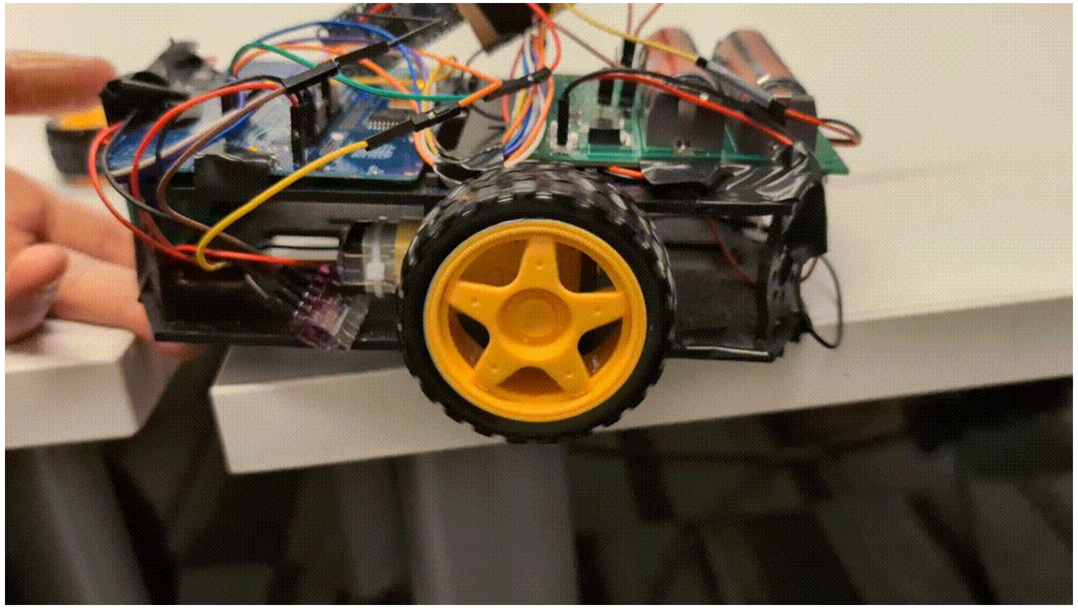
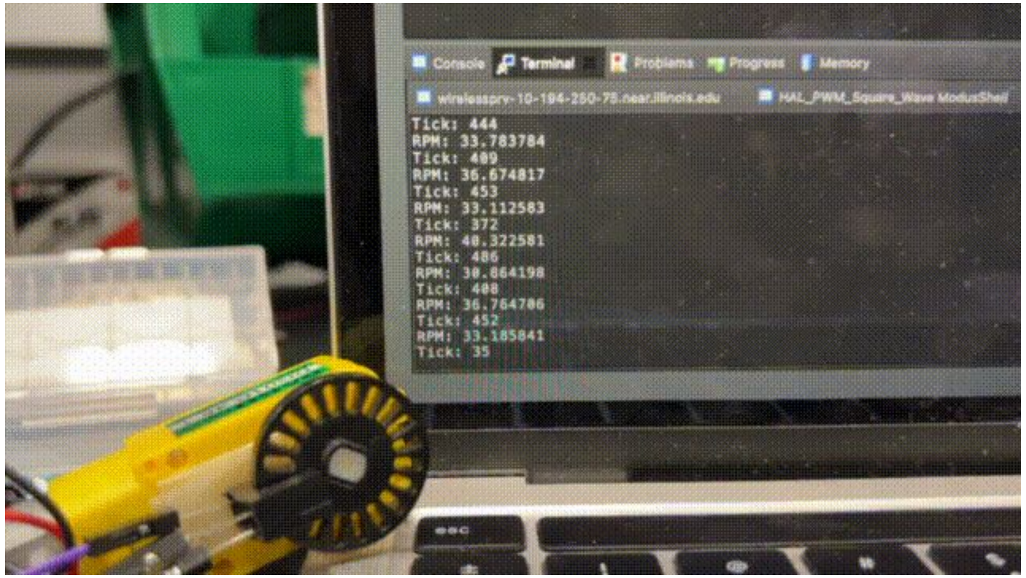


Photo-Interrupter



Measure real time RPM of individual motors

- Each motor has a photo interrupter mounted to it
- Sends analog signal to microcontroller that translates it to digital
- Clock frequency and encoder holes are used to calculate RPM
- RPM used for speed feedback control

Discussion

What were the challenges we encountered during our project?

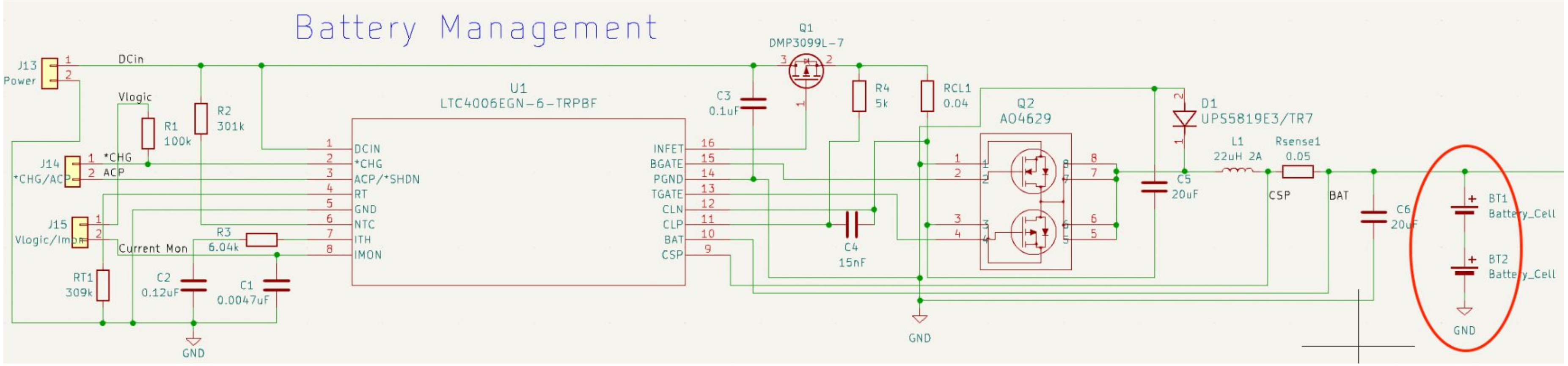
Challenge 1: Battery Management

Removal of Battery Management IC due to safety concerns

- Initially had a charging capability implemented for plug and play functionality
- Decided against it since the IC we chose did not have cell balancing capabilities leading to concerns of imbalance in charge states of the two cells

Solution: Commercial charger

- Used a charger that can charge batteries separately
- Chose batteries with charge protection in built to account for missing battery management IC



Challenge 2: Microphone and Noise

Low accuracy for voice command detection when motor noise is present

- Low accuracy when motor noise is in present
- Motor noise and human voice share the same frequency range (100 - 300Hz)
- Microphones on Dev board are close to motors
- Simple low-pass filters and high-pass filters could not separate motor noise from voice command.

Possible solution: Separation of microphone from the Development board

- Suppress the motor noise and increase voice command signal amplitude
- Relocate the microphones
- External Microphones connected via wires

Challenge 3: Line Following

Line Width and True reverse functionality

- 5 IRs => 3 IRs
- Increase Line Width
- Initial reverse functionality fails to follow line
- Extremely low speeds work 10% of the time
- Due to IR sensors on back end being kicked out when turning

Possible solution: More IR sensors or Turn 180°

- Increase line width
- Adding a copy of the IR sensors to the back of the car
- Turn 180° then go forward

Communication between radar and MCU

- Ideally, we would want a single MCU
- The dev board had microphones for voice commands but not radar
- We had 2 options for the radar:
    - Demo board that send data to PC for processing
    - Wing board with its own MCU for on board processing
- Was not able to perform initialization of both radars through our original MCU

- Hypothesis 1 : Needed way more connections other than the 4 main signals of SPI: CLK,MOSI,MISO,CS → Tested, wasn't enough to resolve
- Hypothesis 2: Connections weren't stable enough

⇒Challenging to establish this many connections stably

Solution:

Used the second option of the wing board with its own MCU, and output a signal to the main dev board

CONCLUSIONS AND FUTURE WORK

What we learnt from working on our project, changes we could make if we were to redesign some parts and, recommendations for further work

Conclusion

- Completed most of the 5 high level requirements

- Radar still picks up some of the things in the background
  - Could be vibration, maybe implement FFT filter
  - Try lower distance and maybe implement direction

- Better speed
  - new motors
  - increase current/number of motor

- Custom printed chassis that has mounting for all electronics

- Include another radar sensor for cruise control

Thank You!

Questions?

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN