EENG383

“Microcomputer Architecture and Interfacing”

Final Project - Anti-Collision Robot

Marvin Ruiz-Ibarra
Robert Metcalf

The Project Idea

Anti-Collision Robot:

- It was decided that our project would consist of a vehicle designed to avoid collisions with objects

- This would be done using a sensor on the vehicle to determine when and object was close by

- Than following a series of parameters an appropriate action would be selected to avoiding colliding with the object

Vehicle Specifications

Object Sensor:

- *The vehicle could sense an object via a number of possible ways.*

  -- IR distance sensor, Sonar

- It was decided that an IR distance sensor would be used on our vehicle

- This was to be mounted on the front of the Vehicle to determine when an object was close by

Vehicle Specifications

Motor Control:

- Two DC motors inside the rotary encoder wheels were to be used to control the vehicle

- Both motors were directly connected to Channel A and B of J7 which is internally hardwired through both H-bridges into the nanocore module

Vehicle Specifications

Power Supply:

- *The vehicle is to be run from a portable 9V battery attached to the vehicle*

- The program was loaded to the board using codewarrior than switched into “run mode” for constant running of the program

IR Sensor Specifications

- The IR distance sensor has a range of between 4 to 30cm.
- Typical response time = 39ms
- Field of View = approximately 45 degrees

IR Sensor Specifications

- *It was noted that if the vehicle was approaching a wall on its side the single front facing IR sensor would not detect this object, therefore it was decided that a second sensor would be placed on the left hand side*

- *In doing this the vehicle would be able to track walls and ensure it remained clear of objects on its left and also directly in front*

- *It would be highly unlikely that the vehicle would encounter an object or wall on both the left and right hand side simultaneously therefore eliminating the need for a third sensor.*

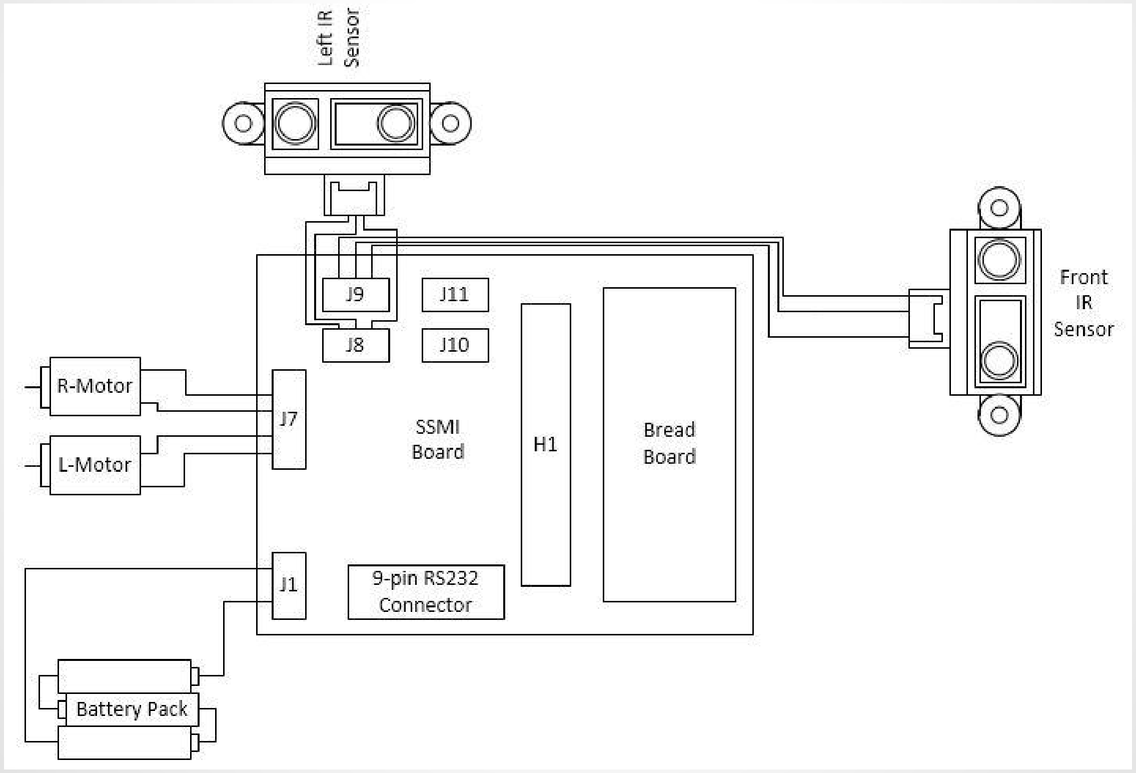
Annotations of Vehicle

- Hardware Set-up

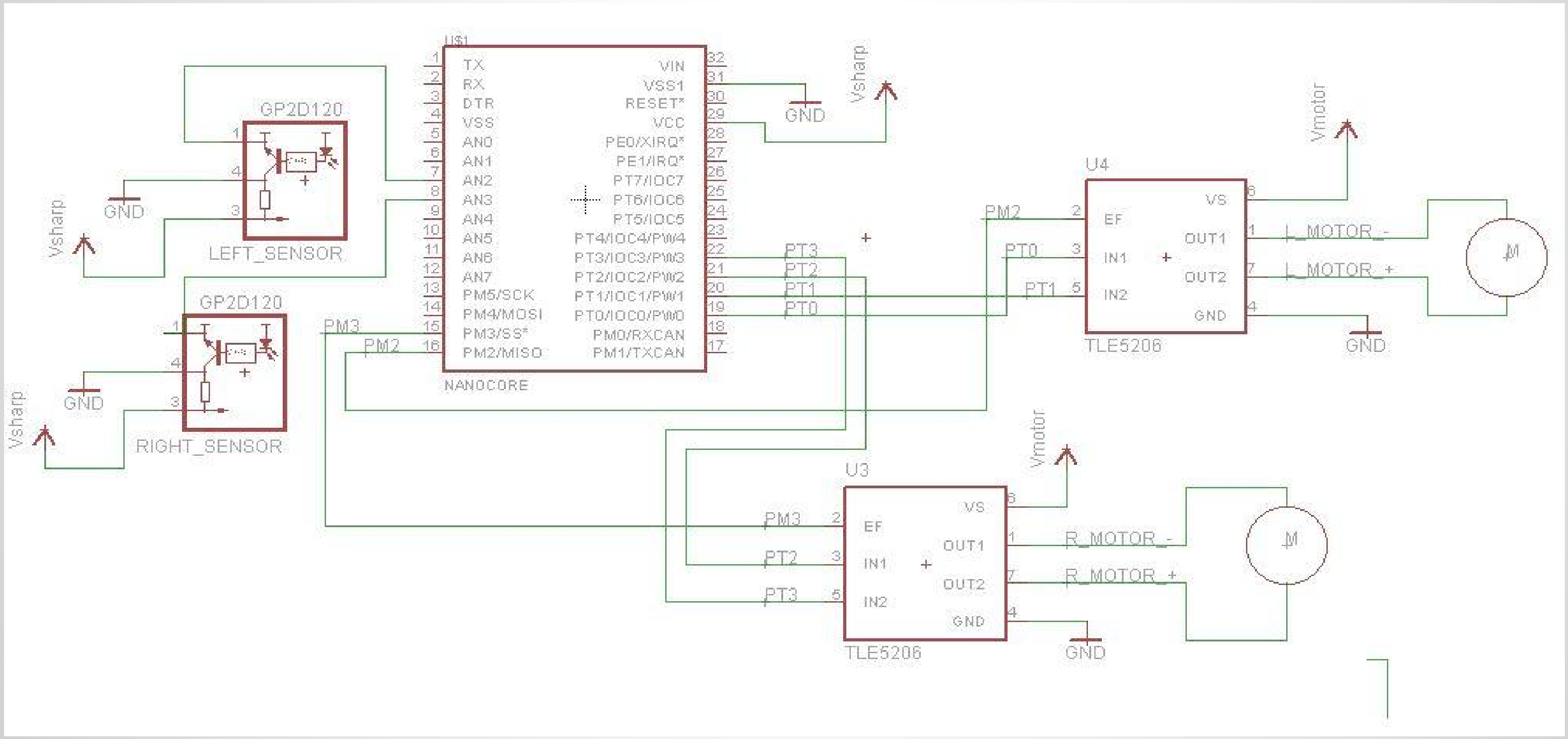
- Final Design Schematic

- Complete Vehicle Chassis with Components

Vehicle Hardware Set-up



Final Design Schematic



The next step - Coding

- Once all design specifications were set the Vehicle was constructed as shown in the previous slide

- It was decided that C-language would be used to program the vehicle

- First of all a rough Pseudo code was drawn up to reflect the required order of operations and instructions of the vehicle

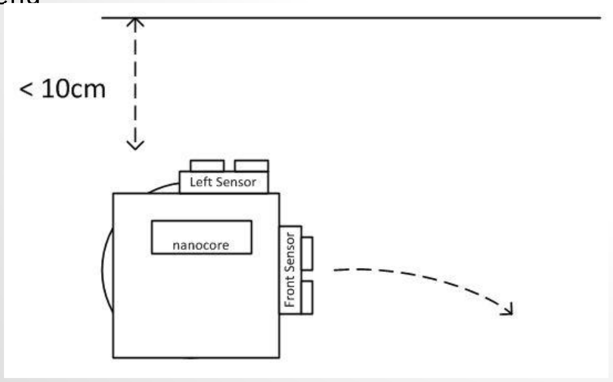
Pseudo C-Language Program

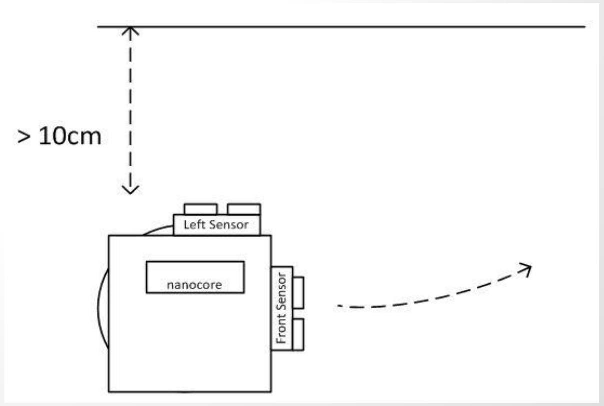
```
Initialize RTI system to time out at regular intervals
  Initialize PT1, PT2, PT3 for digital output
  Initialize ANO2, ANO3, ANO4 for analog input
  Initialize Analog to Digital conversion
  Initialize PWM system clock, and set up PWM to drive PT0 and PT2

Initialize the timer count t = 0
  Initialize STATE to CURV
  while(true)
       STATE CURV:
           if the distance from the left sensor is less than 0xA7 (10cm)
                 drive forward and curve to the right by setting the duty cycle for PT2 with
                 a duty less than 50%
                 Set pins PT0=1, PT1=1, PT3=1
           end
           else if the distance from the left sensor is greater than 0xA7 (10cm)
                 drive forward and curve to the left by setting a duty cycle for PT0 with
                 a duty greater than 50%
                 set pins PT1=1, PT2=1, PT3=1
           end
           else if the front sensor measures a distance of 0xA7 (10cm) or less
           reset timer count t=0
           STATE=TURN
           end
       STATE TURN:
           drive forward and curve sharply to the right by setting the duty cycle for PT2
           with a duty less than 25%
           set pins PT0=1, PT1=1, PT3=1
           if t > tturn
                 reset timer count t=0
                 STATE = CURV
           end
```

CURV STATE

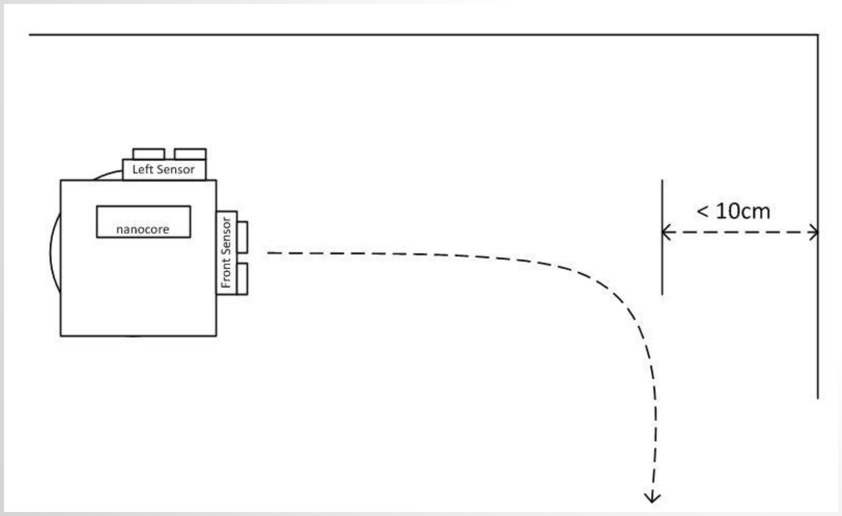
```
STATE CURV:
      if the distance from the left sensor is less than 0xA7 (10cm)
            drive forward and curve to the right by setting the duty cycle for PT2 with
            a duty less than 50%
            Set pins PT0=1, PT1=1, PT3=1
      end
      else if the distance from the left sensor is greater than 0xA7 (10cm)
            drive forward and curve to the left by setting a duty cycle for PT0 with
            a duty greater than 50%
            set pins PT1=1, PT2=1, PT3=1
      end
           else if the front sensor measures a distance of 0xA7 (10cm) or less
           reset timer count t=0
           STATE=TURN
           end
```





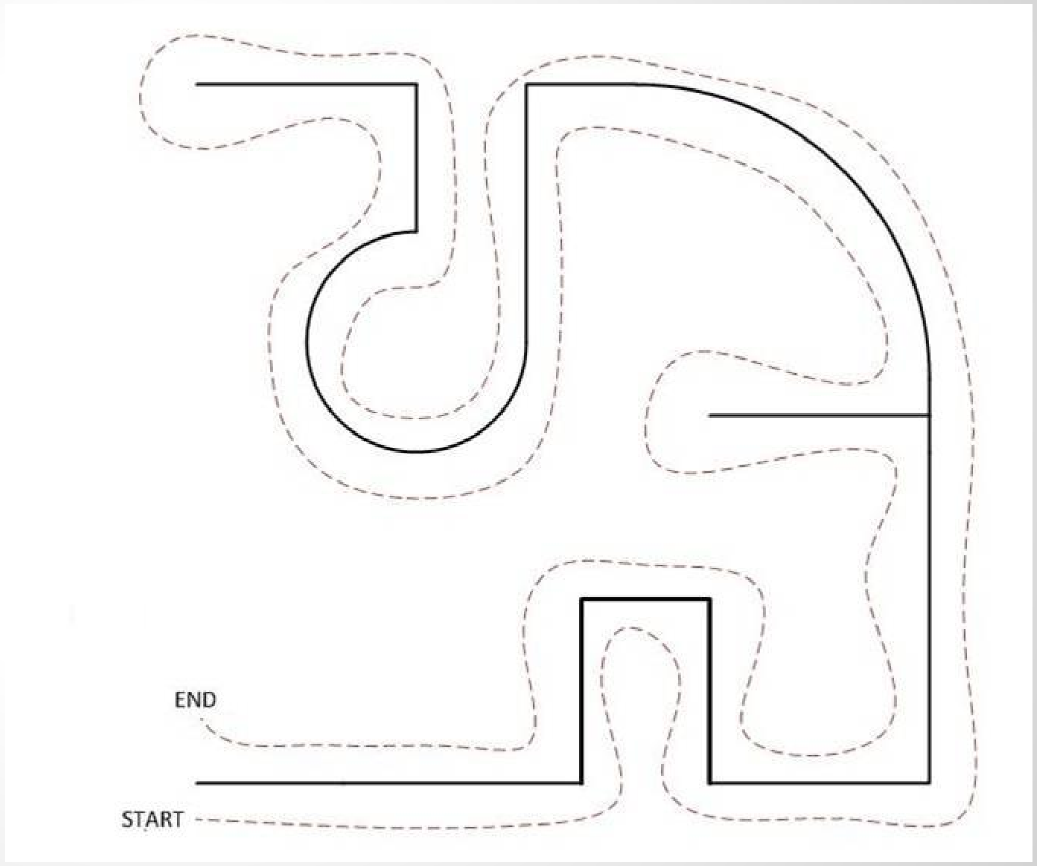
TURN STATE

```
STATE TURN:
      drive forward and curve sharply to the right by setting the duty cycle for PT2
      with a duty less than 25%
      set pins PT0=1, PT1=1, PT3=1
      if t > tturn
            reset timer count t=0
            STATE = CURV
      end
```



Obstacle Course

- Putting our robot to the test
- Walls made out of cardboard



Extensions

- Three or four sensors

in order to detect corners and further improve detection

- Four sensors and motors with omni wheels

move in two dimensions without having to turn

Questions?