

Our top design includes several different parts such as fpga board, slide switches, smart phones, wifi module, two dc motors, dc motor driver, two LEDs, Matlab, Laptop and web server.

**FPGA Board :**

We use the fpga board to read data from UART port that has connected to a wifi module, create PWM signals for driving the dc motors, controlling the speed of the motors, and blinking the lights.

We have two modules in the FPGA. One is motor controller and the other one is the UART module. UART module Motor controller module is responsible of creating PWM signal according to the bits come from the slide switches on the FPGA board. We use only 8 of the switches to set the speed.

**Slide Switches**

Slide switches are used to set the speed.

1 (only the first switch is on) = slowest

…….

256 (eight switches are on)= fastest

**Smart Phones:**

We use one smart phone to stream video from the bot to our laptop. We use an app called IPWebcam. We stream video at 640 x 480. The average frame rate is about 20 frames per second.

We use the other smart phone to remotely control the bot. We use another app called UDP sensor. The app read data from the IMU sensor on the phone and sends it to a device that is connected to a local network. We only read data from gyroscope and send it to laptop.

**Matlab:**

We have two matlab softwares. The first one is responsible of grabbing the video stream that comes from smart phone to laptop, and do image processing on each images. Our algorithm finds out which way to move for Navibot and sends commands to the wifi module on the fpga board. It sends two commands per second.

The second Matlab software is responsible of receiving data from a smartphone that we use for remote control Navibot. It compares the degrees comes from gyroscope sensor with some pre assigned values and creates commands for Navibot. The software sends those commands to the wifi module on the fpga board.

**Web Server**

We wrote a simple web server in Lua that runs on the wifi module. We added some buttons such a blink the lights, reset navibot, turn right, and turn left to do testing our robot. When someone clicks on any buttons it sends a command to the fpga board.

**DC Motors**

We use dc motors to rotote the wheels on navibot.

**DC Motor Controller**

Motor controller drives the dc motors. It is connected to the power supply and fpga board. It receives PWM signals from the fpga board and rotes the motors.

**Power Supply**

We use a light weight 5V usb power bank to power dc motors driver, dc motors and fpga board.

**PicoBlaze**

PicoBlaze is a fully embedded 8-bit RISC microcontroller core optimized for 7-series and older Xilinx FPGA architectures. In this project PicoBlaze KCPSM6 instruction set is used.

In order to compile the software that is written in Assembly code an assembler is needed. The assembler called kcpsm6 is used in this project. It can be found on Xilinx website. After you put kcpsm6.exe and your .psm file in a same folder you simply click on kcpsm6.exe and type the name of the .psm file. It compiles the code and shows if there is any error in your code. Otherwise, it tells you if your code have been successfully compiled and creates .v, .log, .hex, .fmt, log.txt files. In this project we just add the .v file to our source files.

The picoblaze software reads the data on the serial port and compares it with the numbers that represent command numbers. Each command number has a different meaning for Navibot such as blink the right led, blink the left led, or rotate the wheels. If incoming data matches with the command numbers, it sends data to the outports that are used to control DC motors and LED blinkers.