Santiago Gutierrez
kaine@ufl.edu
305-761-0169

John Kurien
jkurien@ufl.edu
954-347-0111

Project Design Report

Project Peek-a-Boo

The purpose of this project was to create a remote-controlled probe that utilized a more intuitive means of control. The controller takes the form of a glove that detects hand movements. These signals are transmitted to the car via a wireless transceiver and the vehicle performs the requested operation. The car also has a mounted video camera enabling it to beam back its location to a remote laptop.

By: Team ICU

Table of Contents

List of Tables and Figures

Tables

Figures

Features and Objectives

The objective of this project was to create an unmanned spy probe. This probe was to be controlled remotely using a more intuitive, non-traditional approach than what is currently offered in the market. Furthermore it was to beam back video of its location to a remote laptop. Features include:

- Intuitive remote control glove
  - Z-axis accelerometer senses hand roll for left/right turns
  - Flex sensor on index finger controls forward acceleration through finger movement
  - Pushbutton on thumb controls backward acceleration
- Zigbee IEEE 802.15.4-based wireless communication system implemented with XBee Pro transceiver
- Streaming video showing exact location of vehicle
- Rechargeable power supplies for control glove and vehicle

The car is able to travel about 300 feet with obstructions and a little over a kilometer line-of-sight. Video is beamed back as long as there is Wi-Fi available.

Analysis of Competing Products

There are no competing products as of yet in the RC vehicle market, although there are a few "wearable electronics" control mechanisms being researched and employed in other areas.

- A researcher at UC Irving, is currently working on a wearable control glove similar to the one in this project that allows him to control a music device while one is performing tasks that require extreme concentration such as skiing and surfing.
- There is a company called ThrustPac that sells a product that provides thrust to push you along on a bike, scooter, etc. using a control glove, although it is wired to the actual product.
- Three researchers from Universidad Autonoma del Estado de Mexico have published a paper on how to telemanipulate a robotic arm through a digital glove.
- A "Fat Controller" snow glove by O'Neill Europe was released in 2005 which controls music devices wirelessly. However, this glove requires the use of the other hand to manipulate a control joy stick sewn onto the top of the glove.

It is evident that none of these research areas or products are directly related to Team ICU's application of a digital control glove.

Concept Selection

The initial motivation for the project was spurred by the Wii Remote for the Nintendo Wii and its applications in areas other than gaming. The innovative motion sensing technology was hailed as a major step forward in gaming and provoked numerous exotic applications unrelated to its original purpose and we hoped to capitalize on a new way to apply it. The team settled on an RC car application because it was seen to have potential in the military or toy market. Controlling a vehicle by hand and finger motions alone is more intuitive than using both hands to control the vehicle. To be able to control the car with one hand, forward throttle was mapped to index finger motions while reverse throttle was mapped to a pushbutton located on the thumb. Turning was mapped to hand motions to take advantage of the hand's natural ability to roll left and right. A wireless camera was added to the vehicle in order to drive the car into areas that otherwise cannot be seen by the operator.

Technology Selection

The integral components of this project include an accelerometer, two MCU's, a wireless transceiver, a flex sensor, and a pushbutton. Below is a brief list of the components required and reasons for choosing a certain product.

Accelerometer

The accelerometer was placed in the glove to detect hand roll. This signals the car to turn a certain direction.

Accelerometer	Cost	Ease of Use	Low Power	Z-axis
ADXL 330	Free	High	340 uA@2.4V	Y
MMA6271QT	Free	Medium	500 uA	Y

Table 1: List of accelerometers considered in project design

When choosing an accelerometer, low power consumption was a big factor. The glove is battery powered so the components should use up as little power as possible. Also, it was important to have an output impedance of less than 32 kΩ in order to properly interface it with the MCU's A/D ports. Although the output can be buffered by a voltage follower, an accelerometer with low output impedance made the glove circuitry design easier and most importantly smaller. An analog accelerometer is preferred for this control application because the continuous reading can be easily sampled by the MSP430's A/D ports. An accelerometer with a Z-axis was chosen because of the hand's natural tendency to rest perpendicular to the earth's surface. The ADXL 330 from Analog Devices satisfied our requirements and has also been shown to work with various TI MCU's.

Transceiver

The transceiver will be used to send control signals to the vehicle from the glove.

Transceiver	Cost	Ease of Use	Configuring Difficulty	Range
XBee Pro	High	High	Low	High
XBee	Medium	High	Low	Low
CC2420	Free	Medium	High	Depends

Table 2: List of transceivers considered in project design

The regular XBee seemed to be a good choice at first, but it was determined that its indoor signal range was subpar. A concern was that the probe would be immobilized as soon as it turned a corner. In choosing the XBee Pro over the Texas Instruments CC2420, the low configuration difficulty as well as its ease of use was taken into consideration. Although free is a powerful motivator to use a certain part, the CC2420 does not come ready to be used outside of the box. The team did not want to have to choose an antenna and other finicky passive components for the CC2420 because it would have detracted from the main goals.

Microcontrollers

One MCU was placed in the glove to interface with the accelerometer, flex sensor, and XBee Pro transmitter. The second MCU was installed on the car to interface with the XBee Pro receiver and control the car's motors. Table 2 shows the available options followed by the reasoning behind the final choice—the Texas Instruments MSP430F1612.

Processor	Cost	Programming	Functionality
Atmel 2560	Free	Medium	Medium-High
PIC	Free	Low	Low
MSP430F1612	**Free**	**Medium**	**High**
MSP430F2013	Free	Medium	Low
Other	>0	High	High

Table 3: List of microcontrollers considered in project design

Initially the MSP430F2013 by Texas Instruments seemed like a good choice. Unfortunately, its functionality was limited by its low pin count. Upon further analysis of the requirements, interfacing our components with an MSP430F2013 cousin—the MSP430F1612—was a better choice because of its high pin count as well as its feature set.

Flex sensor and Pushbutton

A flex sensor was placed on the user's index finger to detect the degree of bending, which correspond to the forward throttle. Likewise a simple pushbutton was placed on the thumb in order to reverse throttle. Research was done on alternatives; however there were no real substitutes. It was important that the device was about 4" in length and thin enough to be taped to the finger of a glove, and that the pushbutton was concealed on the thumb pad. The components used satisfied those two requirements.

Video Camera

The video camera chosen for the project was a generic Wi-Fi enabled camera purchased from Newegg.com. The camera was light enough to be mounted on the vehicle without putting extra strain on the motors. The software to interface the camera with the laptop was also very easy to use.

Flowcharts and Diagrams

Figure 1 below is a block diagram of the components that we will be interfacing for this project. Accelerometer, flex sensor, and pushbutton inputs are fed to the MSP430, which in turn sends the processed signals to the transmitter. The receiver picks up the signal which is sent to the MSP430 on the vehicle. After signal processing, the H-bridges are instructed to activate the motors and thus move the vehicle all the while a video camera is relaying the car location to a remote laptop through Wi-Fi. This information is summarized in Figure 2.

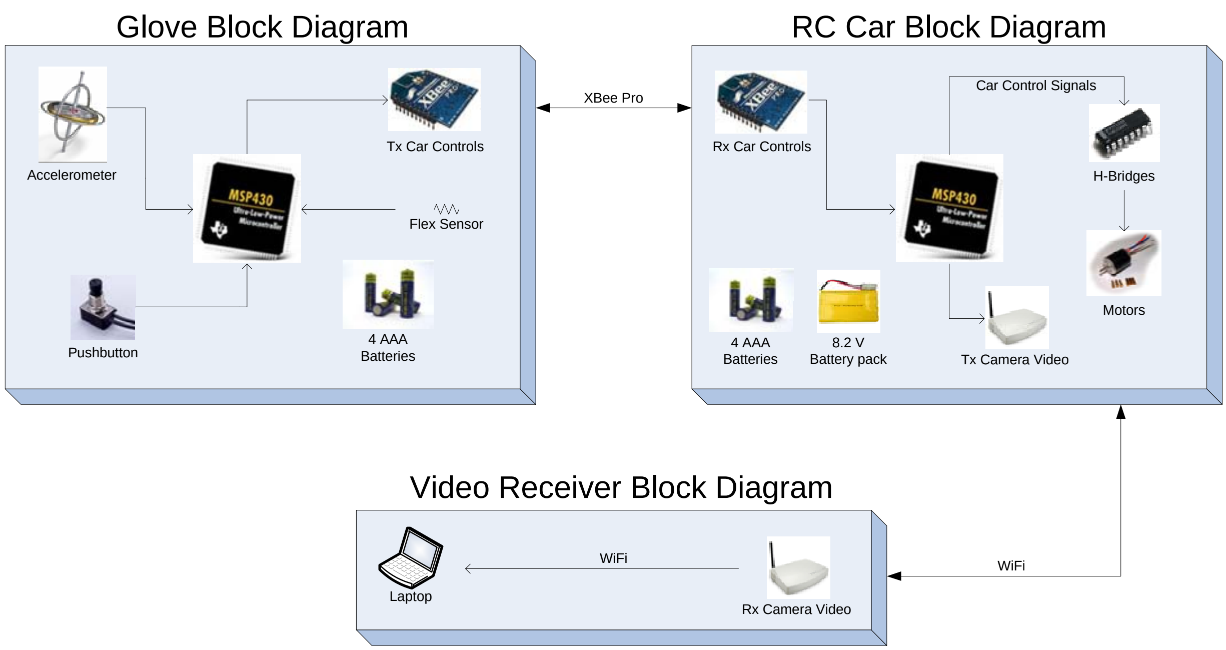

Figure 1: High level system block diagram

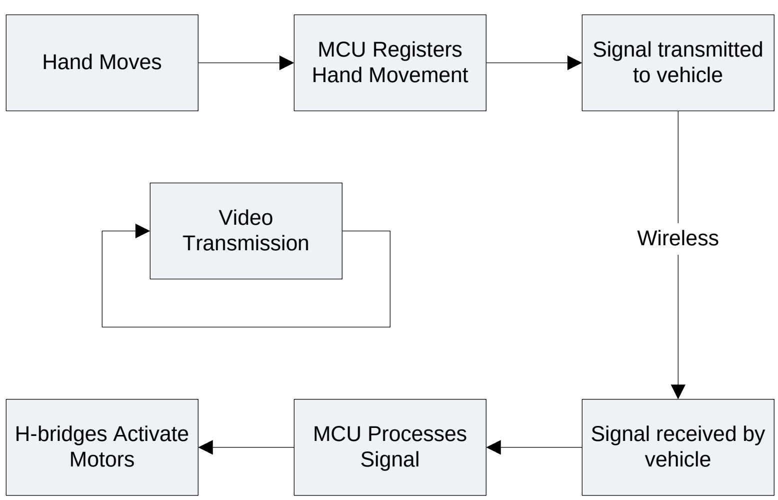

Figure 2: High level system flowchart

Figure 3 demonstrates the software flow for the glove. On power up, the software initializes I/O, variables, ports, the analog to digital converter, Timer A, the USART, and the LCD. It then waits a predefined time for the reference voltage of the ATD to settle. Afterwards, the glove goes into low power mode while it waits for a trigger from Timer A to sample the A/D inputs in the ATD12 interrupt service routine. Once the readings have been made, the digital values of these inputs are compared to preset thresholds in order to decide the car's movement and transmit the encoded information to the car. The ATD sampling is repeated until the user turns the glove off.

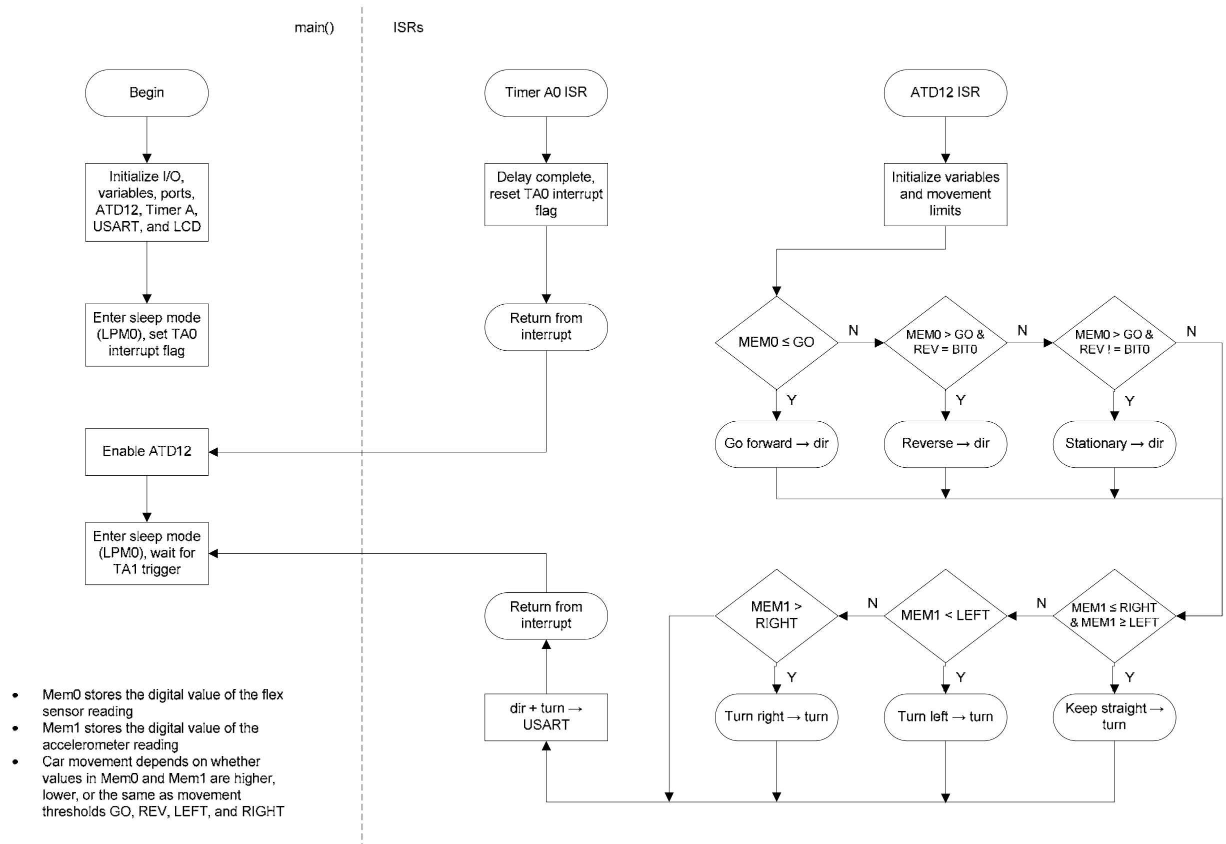

Figure 3 – Glove software flow diagram

Figure 4 demonstrates the software flow for the car. After power up, the I/O, variables, ports, USART, and LCD are initialized and the MCU goes into low power mode. The USART, after receiving data in its buffer, automatically wakes the MSP430 up and fires the USART interrupt service routine. Based on the unique code (rxValue) in the buffer, the proper signals are routed to the H-bridge motor drivers and the car performs a wide range of actions. Following the action, the MCU goes to sleep once more waiting for the USART to trigger it. This process is repeated until power down.

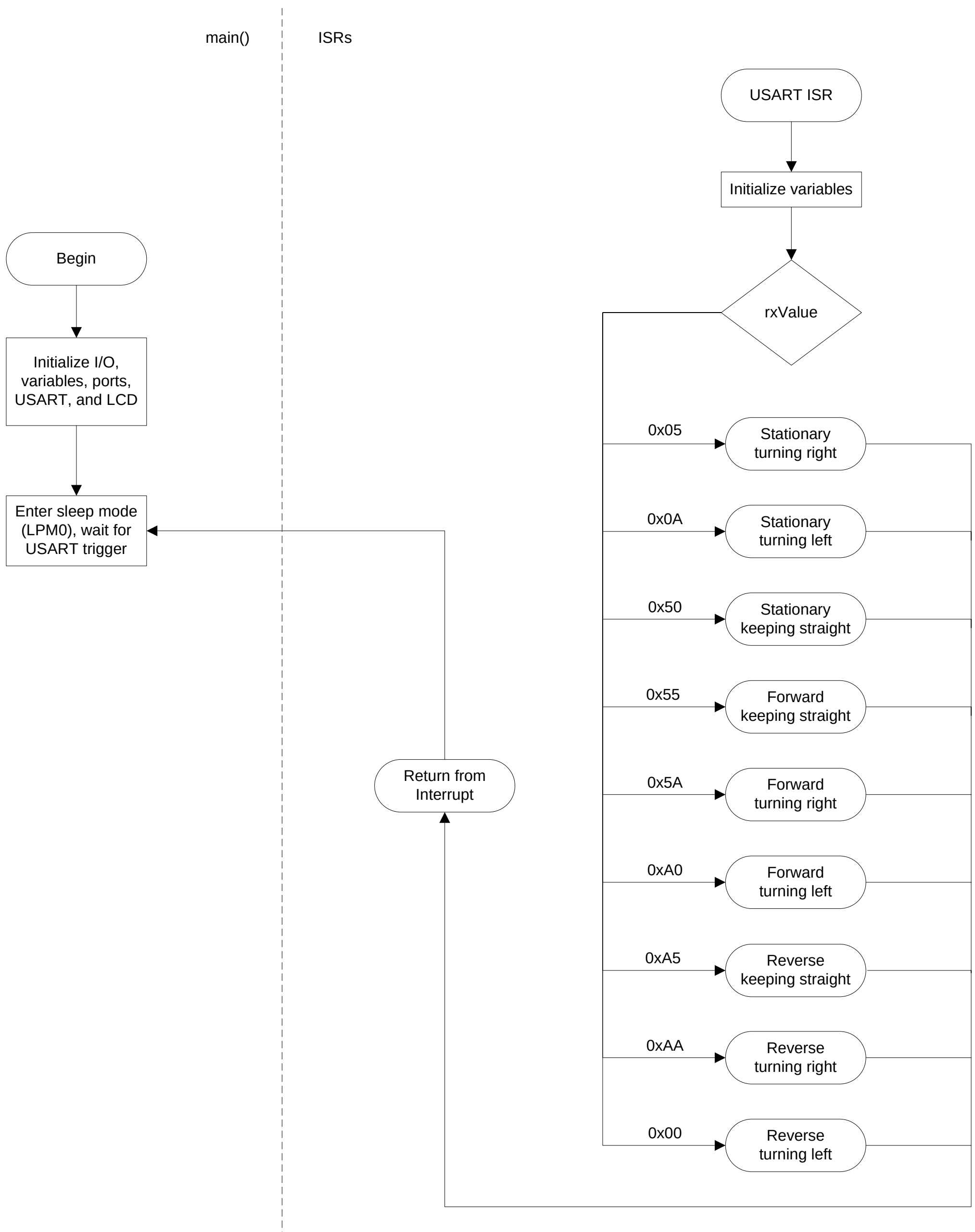

Figure 4 – Car software flow diagram

Gantt Chart and Separation of Duties

Santiago Gutierrez was responsible for getting the accelerometer to function properly. He was also responsible for interfacing it and the flex sensor with the MSP430 as well as determining the necessary power supplies and components. John Kurien was responsible for setting up the XBee transceiver, devising a flex sensor test setup, developing an LCD driver for debugging purposes, and setting up the video camera. The team collectively researched and ordered the parts used, stripped down the car, created a test setup, debugged, and created reports and presentations. This separation of duties is shown in Figure 5.

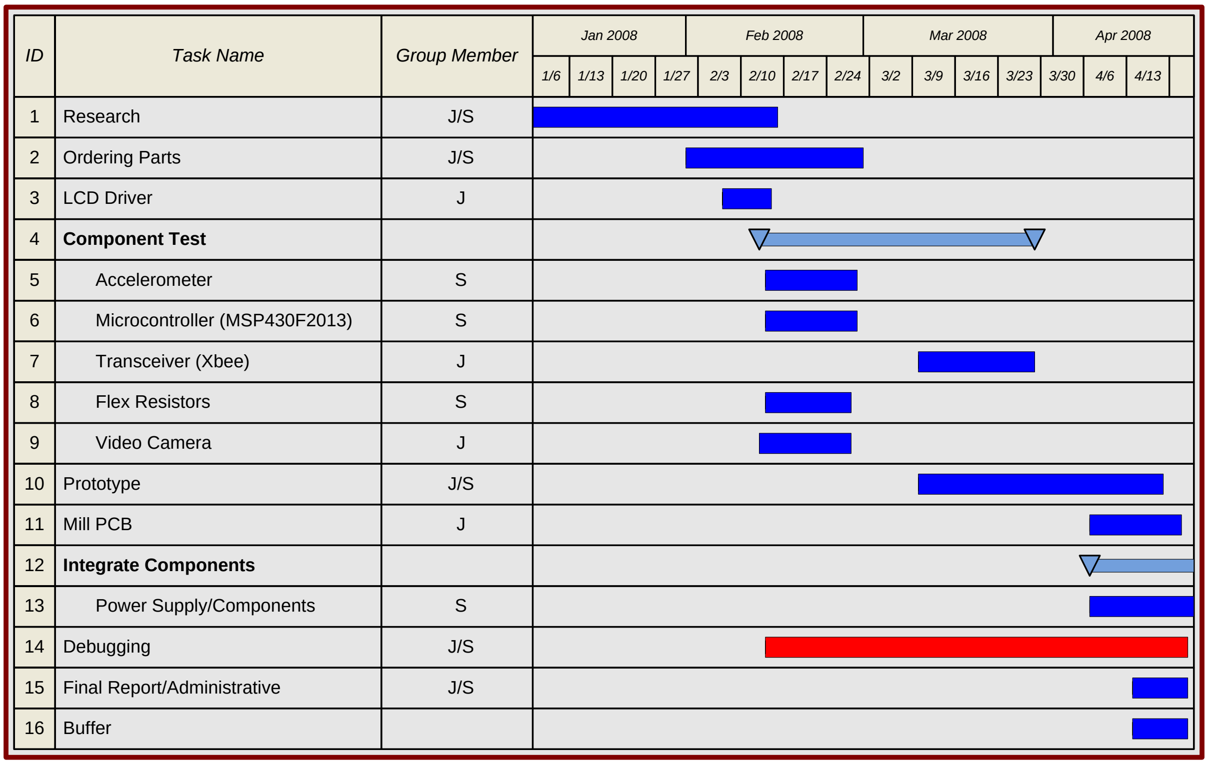

Figure 5 – Gantt chart (J= John Kurien, S= Santiago Gutierrez)

Bill of Materials

Item	Cost
Car	$ 26.27
Camera	$ 86.00
Batt Encl	$ 4.00
XBee pair	$ 70.67
Breakout Board	$ 35.00
Car Battery	$ 35.00
8 AAA Batteries	$ 6.00
Glove/Fabric	$ 5.00
Total Cost	$ 267.94

Appendix A

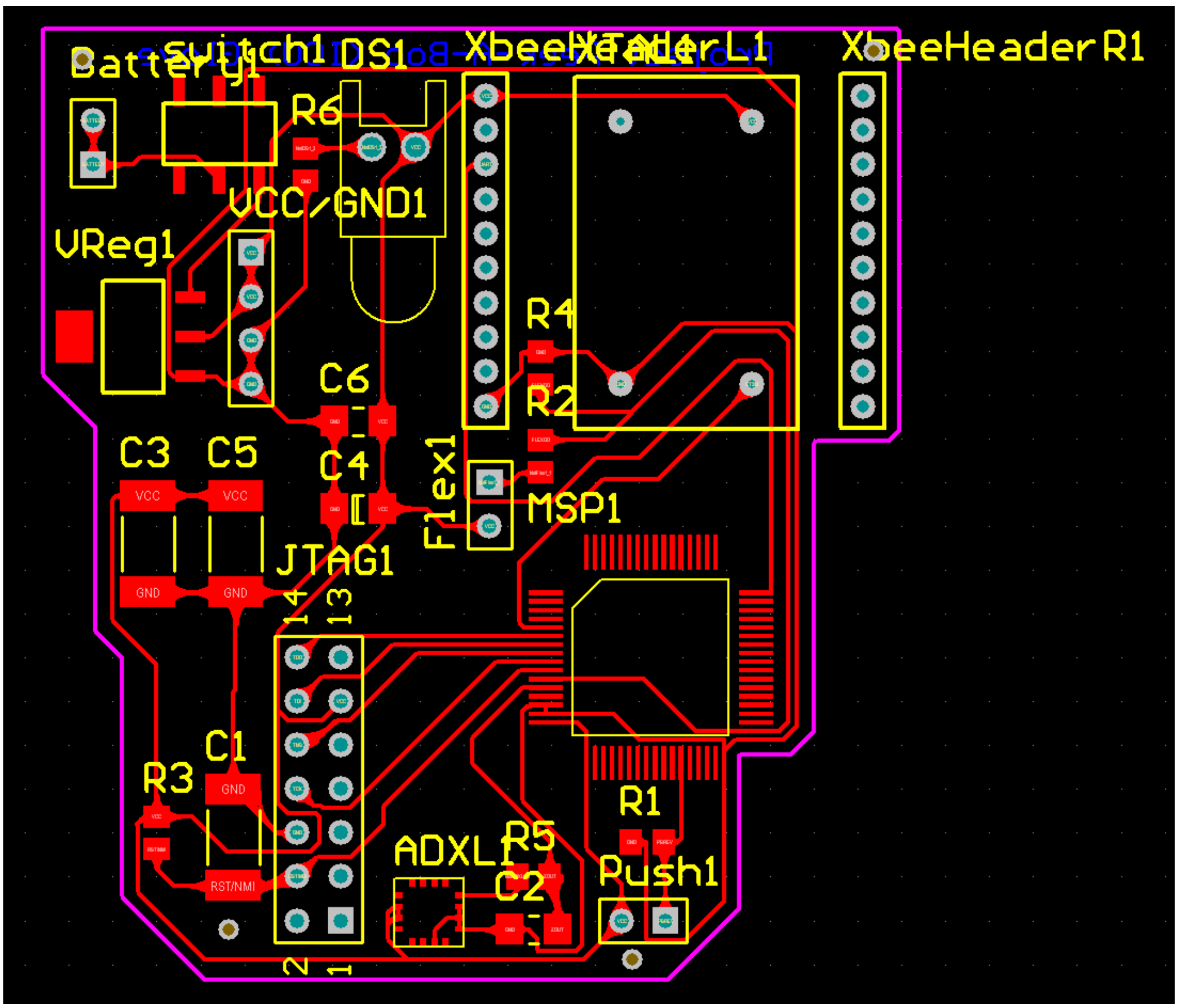

Figure 6 – Vehicle PCB design

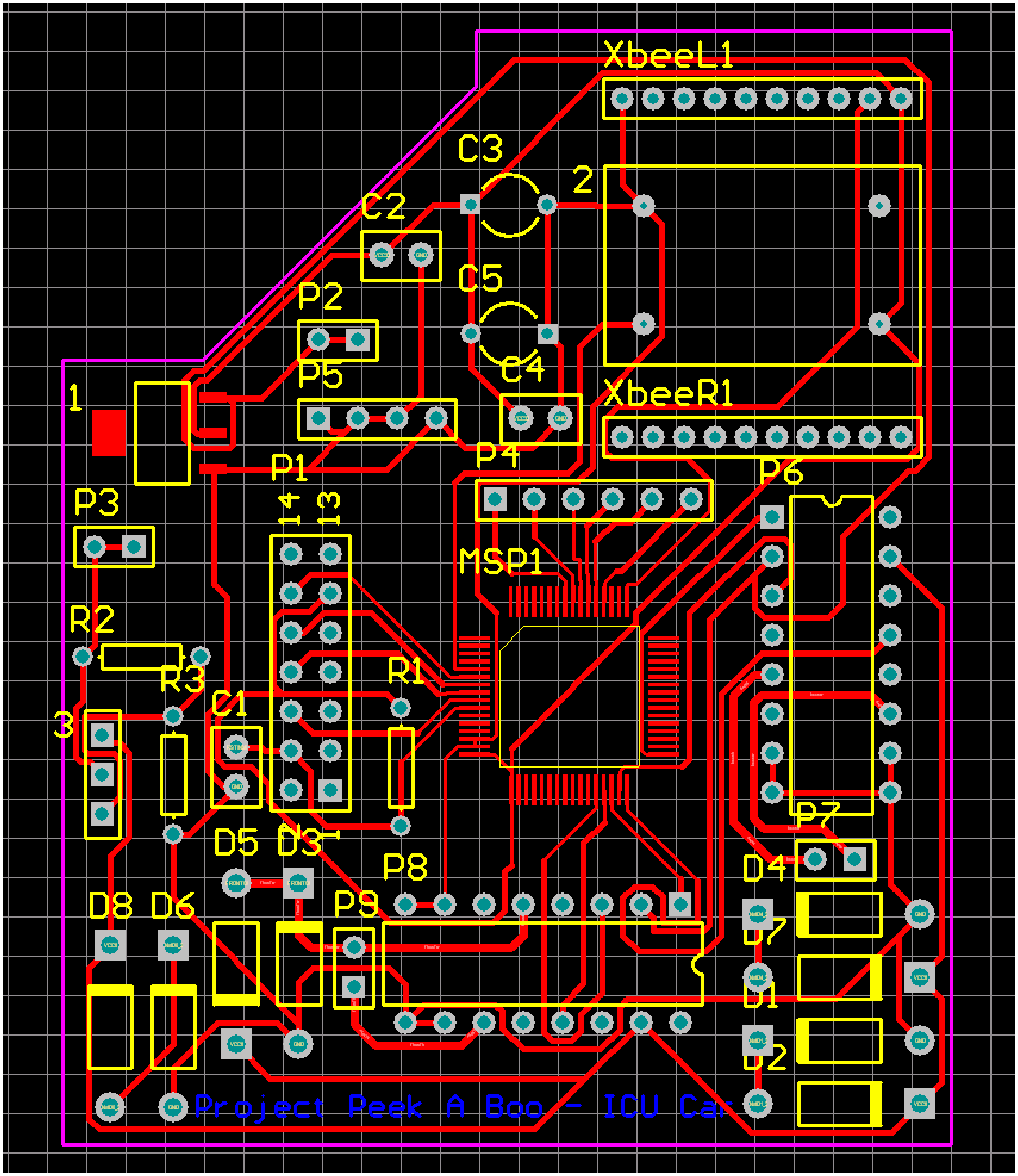

Figure 7 - Glove PCB design