Preliminary Specification

# Platform Supervisor (Linux Host)

The Platform Supervisor will operate on a PC running Linux. Its purpose is to allow a user to control the ERE-EL remotely via Wi-Fi connection.

The Platform Supervisor must run software to perform the following specific functions of the system:

* Communicate with the Raspberry Pi (Linux Box) remotely over Wi-Fi connection
  + 802.11 Wireless LAN using an ad-hoc network
* Provide one of the following, allowing the user to navigate the robot:
  + a real-time keyboard control interface;
  + a real-time joystick control interface
* The selected control interface will allow control of
  + *DC motor speed,*
  + *direction of movement (DC motors),*
  + *camera panning,*
  + *camera tilt,*
  + *any required sensor calibration and control*
* Provide a text display of responses from the platform controller
* Monitor Wi-Fi connection status between the Host and the Raspberry Pi
* Monitor DB9 connection status between the Raspberry Pi and the Platform Controller

The Platform Supervisor may also have a GUI interface to provide a better user experience. This GUI would be written in QT Creator.

# Version Control (GitHub)

To manage source code version control we are using GitHub, a web-based Git repository service. For this project our group has three different repositories for keep things more modular and separated.

Repositories include:

* Documentation <https://github.com/oovi77/Robot-Documentation->
* Linux Supervisor code <https://github.com/kevin-hartwig/Robot_Supervisor>
* Robot Platform code <https://github.com/kevin-hartwig/Robot_Platform>

We choose to use GitHub for its ability to easily manage many files between all group members, track changes made to files, have a full change-log and history of our progression throughout the project.

GitHub is also the more desirable choice for source code version control for this project because it allows us to make all our repositories available to the public (easily providing Bill and Peter access) for viewing but not editing.

# Raspberry Pi (Linux Box)

The Raspberry Pi 3 model B will be used as the platform link system, it will be running Raspbian a Debian based Linux distribution.

The platform will have some way of identifying the status of the environmental logger through the webcam or metal detector.

It will control the HCS12 board using an RS232 link, and will communicate with the platform supervisor using Wi-Fi.

## Hardware specifications:

* A 1.2GHz 64-bit quad-core ARMv8 CPU
* 802.11n Wireless LAN
  + Will be used for communicating with the platform supervisor
* Bluetooth 4.1
* Bluetooth Low Energy (BLE)
* 1GB RAM
* 4 USB ports
* 40 GPIO pins
* Full HDMI port
  + Useful for debugging purposes so we can see the development environment
* Ethernet port
* Combined 3.5mm audio jack and composite video
* Camera interface (CSI)
  + Will investigate this to see if its more viable to use then the camera on the robot
* Display interface (DSI)
* Micro SD card slot (now push-pull rather than push-push)
* Stores the operating system
* VideoCore IV 3D graphics core

# RS-232

RS232 Communication will be used between the COTS system and the platform controller. An external module will need to be added to the raspberry pi 3 so that it can support RS232 communication.

We might need to change the current RS232 communication protocol from last semester so that it can work with an embedded Linux system instead of the platform supervisor running Linux.

RS-232 involves serial communication, communicating one bit at a time at a fast rate. The bits are usually sent in a frame of 8 bits. The speed at which bits are sent is called the baud rate, which is the total number of bits per second.

A general pin layout for RS-232 is as follows:

***Pin 1:*** Data carrier detect

***Pin 2:*** Receives data (this is the pin where information will be sent to)

***Pin 3:*** Transmit data (information will be sent out of this pin)

***Pin 4:*** Data Terminal Ready

***Pin 5:*** Signal Ground

***Pin 6:*** Data set ready

***Pin 7:*** Request to send

***Pin 8:*** Clear to send

***Pin 9:*** Ring indicator

Can use synchronous (program waits for read/write to complete) or asynchronous operations (read/write requests return quickly and program multitasks while waiting)

Operates at voltages of +/- 12 Volts.

# WiFi Sensor Monitoring

Our requirements highlight wireless communication between the Linux host computer and the COTS Linux box (Raspberry Pi 3 Model B).

Our Raspberry Pi 3 Model B WiFi Specs:

* On board 802.11n Wireless LAN
  + Supports 2.4 GHz Frequency Band
  + Supports 5 GHZ Frequency Band

To connect the Raspberry Pi to our Linux host, plan to broadcast commands using an Ad-Hoc Network (aka IBSS – Independent Basic Service Set)

* Socket to socket communications
  + Each node acts as a transmitter and receiver
* No existing hardware required (WAP, Routers)
* Flooding algorithm for forwarding data
  + Every packet is sent through every outgoing link, except the one it arrived from

[Link for Reference](https://spin.atomicobject.com/2013/04/22/raspberry-pi-wireless-communication/)

# Environmental Logger Sensors

## Thermometric Relative Humidity Sensor

Polymer based relative humidity sensor for humidity monitors and controllers, air conditioners, humidifiers and dehumidifiers, automatic ventilation.

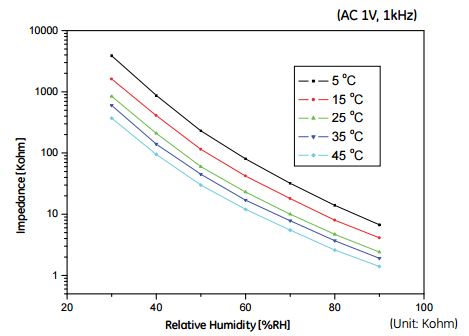
Relevant Electrical Specifications:

Rated Working: Voltage: AC 1 V (50 Hz ~ 1 kHz)

Rated Power: 0.3 mW

Nominal Impedance: 60 kΩ ± 30 kΩ

Typical Humidity Curve



**Pressure Sensor**

* Found most to read 300-1100hPa (4.35-15.95psi)
* Most have I2C interfacing andMost come with amplifying circuits

Barometer with unamplified digital outputs:

[Link to Sensor and Datasheet](http://www.servoflo.com/sensors-by-application/barometric-pressure-measurement-a-compensation/item/463-ms5607.html?utm_source=EMS+Applications+-+July+2013&utm_campaign=EMS-Newsletter&utm_medium=archive)

Requires:

* Voltage stepping with OP-amps
* Logic to convert data into something meaningful

Features:

* Digital outputs
* I2C interfacing and SPI interfacing up to 20MHz

**Analog Output Inductive Proximity Sensor (Metal Detector)**

Description of how sensor works: Sensor produces an electromagnetic field through oscillation, when metal approaches the field the circuit closes and current flows through the sensor changing the voltage level. This can be integrated with a Schmitt trigger to detect metal.

-fargo controls inc, sensor range: (7.3 – 22mm)

Outputs: 0 – 10V based on the distance between metal and the sensor.

Power supply: 18-30V

size of sensor: (45mm by 55 mm)

<http://www.globalspec.com/pdf/viewpdf?partId>={B16D88EA-D3F4-4880-9C83-047B38030952}&comp=52&from=detail&vid=96420

<http://www.digikey.ca/product-detail/en/panasonic-industrial-automation-sales/GX-F6A/1110-2211-ND/3899739>

Some of the sensors I've found need 12 to 24V, and their sensing distance is a couple mm. This could be incorporated into the project, not sure if we can use the 24V from our board or if we'll need another power supply.

**Potential Outline for the report**

1. Sensor introduction and description
2. Reason for putting the sensor in the project
3. Sample circuit for signal conditioning
4. Conclusion