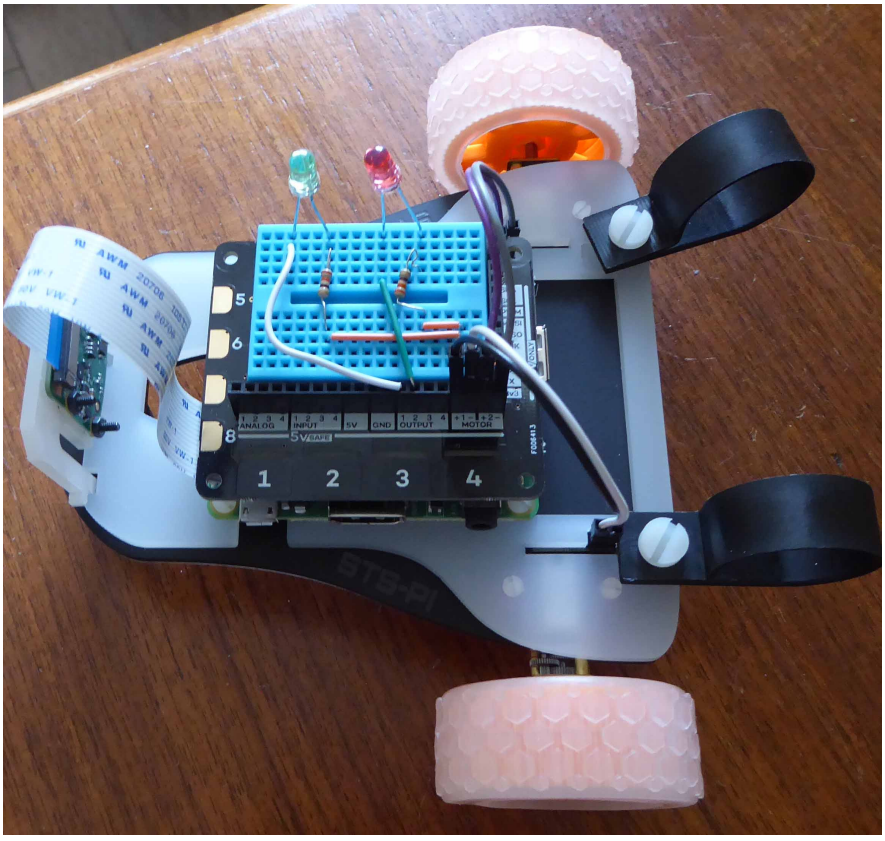
Almost every car and dwelling has a security system where you must enter a code before you can enter. I decided to add a pin entry system to the latest robot. You have to enter a four digit code and you get three attempts before the system shuts down. Red light means bad code and green light means correct code. Since I am using the digits 1-4, on the breadboard, that means there are 4 x 4 x 4 x 4 = 256 combinations.

Who remembers their high school science class about volts, amps, ohms and how they are related? The LED lights are very sensitive and I have to ensure I do not blow them up with too much volts/amps (I have choice of 3.3 or 5V) and therefore must select the correct resistor. Following the measure twice and cut once rule, I will start with 3.3V and the biggest ohm resistor and decrease the ohms until the LED lights up.

The resistors look like a single piece of brown rice with a wire at each end. The resistors are color coded, multiple bands, to indicate the ohm value. Interpreting that code took two days of research. My resistor pouch has 500 pieces with 20 different ohm values.

So this is how it looks on the new robot, Basic Research Instrumentation Explorer (BRIE):



And here is the video of using the pin entry system:

http://systemkeepers.com/robotics/v/pin-entry.mp4

Any one who has seen any of the Toy Story movies, knows what happens when humans are not around. I have often wondered what happens with our robots when I am not around. According to the inventor of the iroomba robot floor cleaner, the problem that many robot manufacturers have is that they try to have the robot accomplish too much. The idea is that you increase efficiency when you make something more simple.

If you think of a construction site, you have the basic labourer, the electrician, the carpenter, other trades and then the site foreman/forewoman/foreperson. The foreperson directs the others on what work to conduct.

So our latest project is to have two or more robots be able to communicate/talk to each other. We had done a simple version of this with Allie. We used an USB cable to connect the Raspberry PI computer to the Arduino processor and then sent serial commands between the two. The Raspberry PI computer acts like the foreperson. The Arduino processor has the sensors/motors attached to it and acts like the trades person. However this would mean that all our robots would have to have a cable attached to them. It would look like the daycare kids going for a walk, all holding on to a rope.

Fortunately, all of our Raspberry PI single board computers (SBC) include a wifi capability. Therefore we can assign a unique address to each of them which is called an Internet Protocol (IP) address. We will use a communication protocol called User Datagram Protocol (UDP). Datagram is just a fancy word for messages.

Just like our construction site has a foreperson and multiple trades people, UDP has a server and multiple clients. Another similarity is that, with UDP, there is no guarantee that a message/order sent will be received. “But boss, you never told me that!!”.

Note: I told an ex-employee/friend what I was going to do. He makes his living programming embedded systems, where the average salary is $140K/year. He warned me “remember the robots are watching and just waiting for you to make a mistake”.

Say you have now programmed your car to have two way communications with a mapping program. You are merrily driving down a road and get to the end where there is a deep canyon. The mapping program tells you to stop, do an 180 and the drive forward. However, since you are using UDP, you do not receive the “do an 180” command and just the stop and drive forward commands. Oops!!

Transmission Control Protocol *(*TCP*)* is one step up from UDP in that it establishes a connection between a sender *and* receiver before data can be sent. It also numbers the transmissions so that they are received in the correct order. However, as any partner will tell you, a response of “uh-huh” in a long discussion, is NOT much of an acknowledgment of understanding what was said.

Now that I have two machines (ROJO and BRIE) talking to each other, what practical stuff can I program them to do? Of course “practical” is different when one is in a lab. The criteria are “challenging”, “will not blow up”, “will not raise blood pressure toooo much” and only going a maximum of 50% over budget.

I have always been interested in the remote control robots for bomb detection and searching other unsafe areas. However most of them use a special dedicated control box. I decided that any machine should be able to act as the controller/server. The first piece of software is remote control of BRIE. I press keys on ROJO that give BRIE directions:

	W			8			forward	
A	S	D	OR	4	5	6	means	left stop right
	Z			2			backwards	

+ increase speed - decrease speed C camera snapshot

Initial testing was done on the main level floors. That stopped after receiving the evil eye from the CFO. Subsequent testing was done in the lower level on rugs.

So here is a video of BRIE searching the lower level main room.

http://systemkeepers.com/robotics/v/remote-control.mp4