Intro

Background

Dissecting the original controller

Connecting a Raspberry Pi…

Software notes & links

Upgrading the chassis

Near final assembly

Next plan…

The Raspberry Pi controlled go kart ‘tug’ has evolved over time. At first, BigBot, as it became known, was going to be a Dalek as the drive mechanism is pretty much perfectly sized to form the base of a full-size Raspberry Pi controlled Dalek. But projects like that take months of work so step 1 was to just get a Raspberry Pi to control it and see what happens next. Another idea was a Garden Explorer robot, something a little more rugged and fitted with a camera that could navigate a garden, possibly even mow the lawn... But there’s only so much time to do these things. Then one day shortly after replacing the aging battery on BigBot the with a brand new car battery, the youngest daughter came up to me to ask her to pull her around the garden on the go kart, so that started the next crazy idea…

BigBot started out in life as an electric golf trolley. It was originally controlled by a thumb-wheel control near the handle that you turn to make it go forwards at different speeds while you steered it the old fashioned way using your hands. But it also had a radio controlled mode where you could make it go forwards, backwards, left and right using a five button remote control. The fifth button was the all important stop. Having a golf trolley run into you tends to hurt a bit so a stop button makes sense for these things. It also helps to reduce the number of times it falls into the water hazard ad the local golf course… Thankfully this one has remained dry so far…

Pic of original control

The radio receiver appeared to be a plug-on option. A 7 pin connector joined the radio receiver to the main speed controller micro. So the thought was that it should be possible to remove the radio receiver and replace this with a Raspberry Pi. With a little trial and error the 7-pin connector revealed it’s function. 5V power, ground and a 5-bit control function that controlled the direction of motion plus speed. There was also a remote controlled horn but I was never able to get that to work.

The next step was to get the Raspberry Pi controlling the 5-bit control to the motor controller. The motor controller on the golf trolley was 5V so the 3.3V outputs from the Raspberry Pi needed level-shifting from 3.3V to 5V. One of the many drawers in the workshop is full to the brim of 2N2222 NPN transistors which are perfect for this sort of thing. So five transistor based (inverting) level shifters were quickly assembled onto a Proto-Zero board that I’d kept for just such an application.

Schematic of assembled Proto-Zero board

The next step was to write the software. Thankfully, from running the MicroPiNoon robots and other robots for the Cotswold Raspberry Jam, I had software which was mostly complete. All I needed to do was to replace the functions that usually control the Edukit3 compatible PiZ-Moto pHat with functions to drive the 5-bit output that goes to the golf trolley’s motor controller via the level-shifter. Remembering to account for the inversion…

Software available on github…

Testing went well with everything working. At this point it was seen in the wild at Raspberry Fields 2018 and a few people got to play with it. At this point, it’s worth pointing out the problem with the Rock-Candy wireless PS3 controllers – after a few minutes of no-activity they shut down, ready to be woken up again. On waking they search for something to talk to but this isn’t necessarily the thing they were talking to previously. For example, someone picked up the control for the golf trolley at Raspberry Fields and found it didn’t respond. What did respond was the winner of the Intermediate class of PiWars 2018 as the amazing X-Bot switched itself to listen to the Rock-Candy controller for BigBot and proceeded to drive itself off the table with a sickening crash…

The next challenge was