**PLANS AND PROGRESS**

The Socapel servo motor drive data reader/loader began through the frustrations of the maintenance team at newscorp Australia, the team were not only becoming frustrated with unreliable equipment that would lose power and drop out half way through programming procedure, they were also getting frustrated by the time it was taking to program the socapel servo motor drive.

The original programming procedure used a very basic manual entry keyboard that you would have to enter commands one at a time, check the data value, correct it if needed then progress to the next until all 255 parameters were correct or corrected.

e.g.

1. Check equipment label
2. Find reference page in book
3. Enter first parameter – **ME034**
4. Press **ENTER**
5. Check data value and reference to book

IF DATA IS INCORRECT

1. Type **n** followed by correct value (ie n0000) then press **ENTER**
2. Type **I** then **ENTER** to increment to next parameter

IF DATA IS CORRECT

6. Type **I** then **ENTER** to increment to next parameter

.

.

Continue this cycle to all parameters are checked and corrected

.

.

If all parameters are correct enter **S** then **ENTER** to save all data to socapel drive

The first advancement was to use a laptop, testing was done through the program **terminal,** this terminal program was a slight improvement over the manual keyboard as it would not drop out as did the original programming keyboard. The one draw back was that anytime you needed to change a data value you would have to edit the text file.

The second idea was to develop software on a laptop based on the python3 programming language, the python3 programming language had the required GUI module tkinter that was built in, we then looked for communication modules that could be added to the python programming language that would assist with the ASCII RS232 communications, we done our research and come across an addon module called pyserial, pyserial had a free software license that would allow us to use it and was designed for ASCII communications. Some work and investigation had been done around this idea, the one wall that kept coming up was being able to leave the laptop in the required area and having it secure, this raised concerns and was a major problem, so the group put their heads together to come up with a better solution.

The third idea was to develop something much more portable, smaller than a laptop, simple to use, rechargeable and could be left in the area that required it, would not be needed else ware and the security concerns of a laptop would not be an issue. The team come up with a raspberry pi based data loader, it could still run the python software, it could have a seven inch touch screen, add a portable rechargeable power supply, and would be very portable while removing the risks of a laptop.

Currently the team is in between the second and third idea, there was some work completed on the laptop idea and a basic GUI was being developed, the team is still looking at improving the GUI and giving it a more pleasurable appearance , with the team moving to the idea of a raspberry pi based module it brings changes, the team must now work on and investigate the differences in programming for the rs232 serial communications, and also the file transfer procedures for storage of the programming files. The team is also looking at the best products and solution based around a rechargeable power supply that is small, long lasting and portable enough to ensure that the raspberry pi will continue to have power for as long as possible, so it requires minimal charge.

The team has also discussed its next advancement after the raspberry pi unit is complete, and would like to possibly develop an android and iPhone app. this would require the team learning these same steps such as GUI development, serial communications and file manipulation based on the java and swift programming languages.