2011 SuperDARN Workshop

Dartmouth College, New Hampshire, USA

May 30 - June 3

New Operating System Upgrade for the Bruny Island Radar

Kamalakkanan K., Vu Q., Greenwood R., Custovic E. & Devlin J.

Department of Electronic Engineering

ABSTRACT

This poster presents recent work on the installation and testing of the current release Radar Operating System (ROS) for the Tasman International Geospace Environment Radar (TIGER) at Bruny Island.

CURRENT SYSTEM

Current the Bruny Island ROS uses a QNX operating system to generate the timing sequence. New Linux based ROS available at Superdarn website provides a good alternative for existing QNX based ROS. Linux is free and a reliable general purpose operating system but lacks the real-time capabilities of QNX necessary for RADAR control, especially for the generation of the timing sequence for TX pulse, TX/RX envelope and sampling trigger.

The timing sequence can be generated by a low cost microcontroller/FPGA add-on board but involves considerable time and cost.

The real-time bandwidth required for the generation of the timing sequence is in the order of a few kHz which could be easily met with any general purpose PC provided the operating system is able to meet the real-time constraints. The solution is to use freely available real-time extensions to Linux kernel which would then provide the necessary real-time bandwidth for the generation of the timing sequences.

The current implementation involves RTAI(Real Time Application Interface for Linux) for reliable and easy implementation of real-time tasks for RADAR control.

TIMING COMPUTER

The replacement timing computer uses Linux/RTAI real-time scheduler for generating the timing sequences. It uses most of the old timing computer code to generate the timing sequence and control the radar hardware by utilising PCI 8255/8254 parallel interface cards. The timing sequence is generated from a tightly controlled real-time process which has higher priority than any other process and hogs the hardware until it has completed generating the timing sequence and yields the hardware to lower priority tasks.

The timing computer uses an old Pentium PC and meets the real-time constraints in spite of running an X window and other heavy Linux tasks.

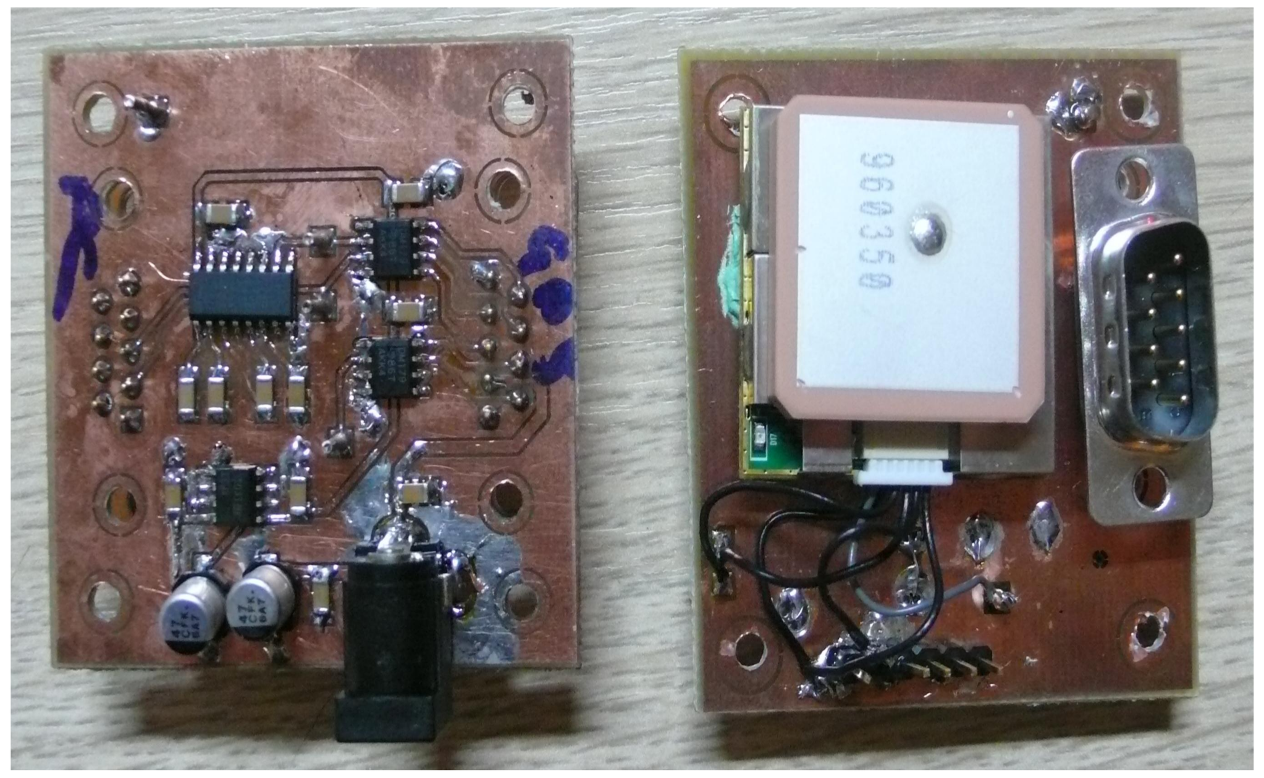

Figure 1: In house designed GPS module unit

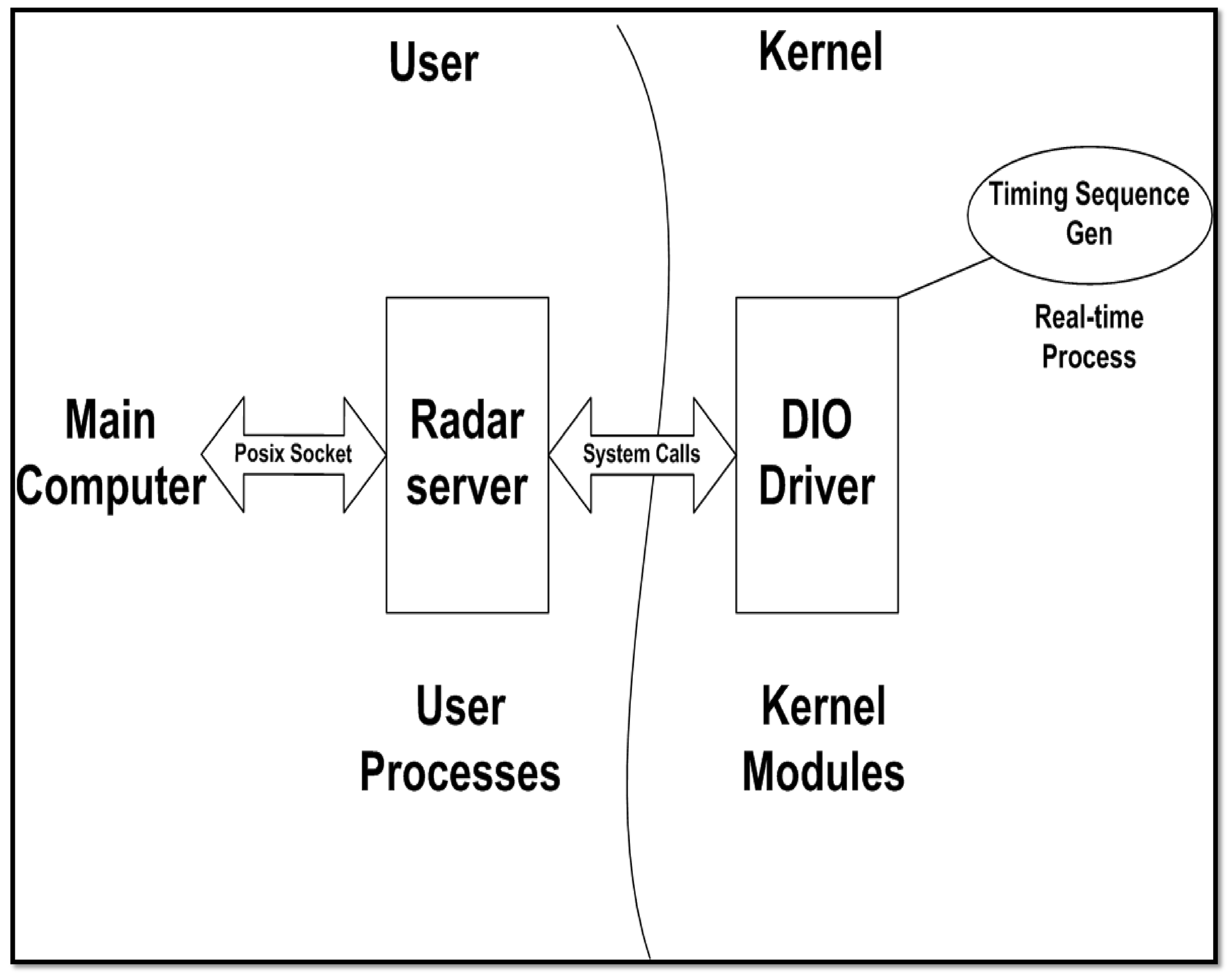

Figure 2: Generating timing sequences by Linux/RTAI real-time scheduler

All User Processes are time sliced by Linux scheduler while real-time process are scheduled by RTAI. By yielding the control to Linux during non transmit we can use all functionalities Linux provides and at the same time makes sure hardware is not taken away by Linux during critical times.

MAIN COMPUTER

ANALOG TO DIGITAL CARD

The replacement main computer runs a normal Linux distribution and Superdarn-ros.3.1 beta operating system. A new A/D card is to be deployed to provide longer buffers and additional functionalities including many different modes of A/D triggering which could collect additional samples. This is convenient for special LTU modes and collection of extra data, during normal

Figure 3: Data Translation DT301 AD card with PCI interface

GPS

Current Bruny Island computer has an ISA GPS card (Off the shelf PCI GPS card are very expensive). A new yet simple RS232 interface with EM406A GPS module provides a cost effective solution to synchronisation (Figure 1)

CONCLUSION

We have developed and demonstrated a cost effective alternative for an embedded computer for the Linux based radar server architecture. The overall cost of replacement of the ISA based hardware is quiet low. Existing PCI based A/D cards could have very well sufficed for normal functioning of RADAR. Any old PC could reliably generate the timing sequence and are easily adaptable. Dummy tests have shown that the performance meets the requirements and is awaiting final installation at Bruny.