Hi Freaksl.

Whilst working with the reversed-engineered AVR based Mobile listed Here,

I thought it would be cool to interface a Tiny13 to the Host AVR (8515 look alike). I figured that the inclusion of a smart A2D would be a useful expansion of the already plentiful on board capabilities.

Having firstly considered installing the Tiny13 as a software I2C Slave peripheral @ Address $00/$01, I considered that it might be possible to communicate with the Host AVR using the already to hand SIM Card Interface, by gently soldering the connecting wires to the SIM Socket.

Consulting the relevant **ISO7816** specs. it became clear that the Tiny13 could more than handle the job and the attached programs document these efforts.

Simplex.asm :

This is a fully sofware driven serial , Interrupt free comms program, devised to test the feasability of the Virtual Sim Card approach. In particular, I wanted to check how easy it would be to generate the necessary Parity bit in Software. The result speaks for itself !

Accu.asm:

This is where the action starts. It consists of basically the same logical program as the Simplex.asm application, suitably modified to allow for a higher thruput.

I wanted to monitor the onboard Accumulator using the Tiny13 10 bit A2D and feed the processed result to thr Host Avr via the Sim Card Bus. This neccessitates reading the A2D 256 times a second and averaging/ Low pass filtering the 256 decimated 8 bit samples Then the Tiny13 is put to Sleep in the power down mode in order to conserve power.

Once a second the host sends a Wake Up command (**$FF**) to the Tiny13 via the Sim bus and requests the 1Byte Voltage data.The receive Command byte with positive parity is sensed when INT0 goes Low and the Tiny13 awakes. The Accu voltage must first be stepped down from Max 6Volt to Max1.1Volt using a simple Resistor divider. This is necessary because we are using the onboard 1.1Volt reference voltage

The A2D outputs a Digital representation of an analog value according to the following equation:

Value = Vln * 1024 / Vref

In the case of a normal NIMH accu we could expect a VOut, depending on the accu condition, of between 2 to 6 Volts. After steeping this voltage down thru the resistor divider, we could expect to obtain values for Vin lying between the extremes of .4 to 1.1 Volts. And of course we've chosen Vref to be 1.1Volt.

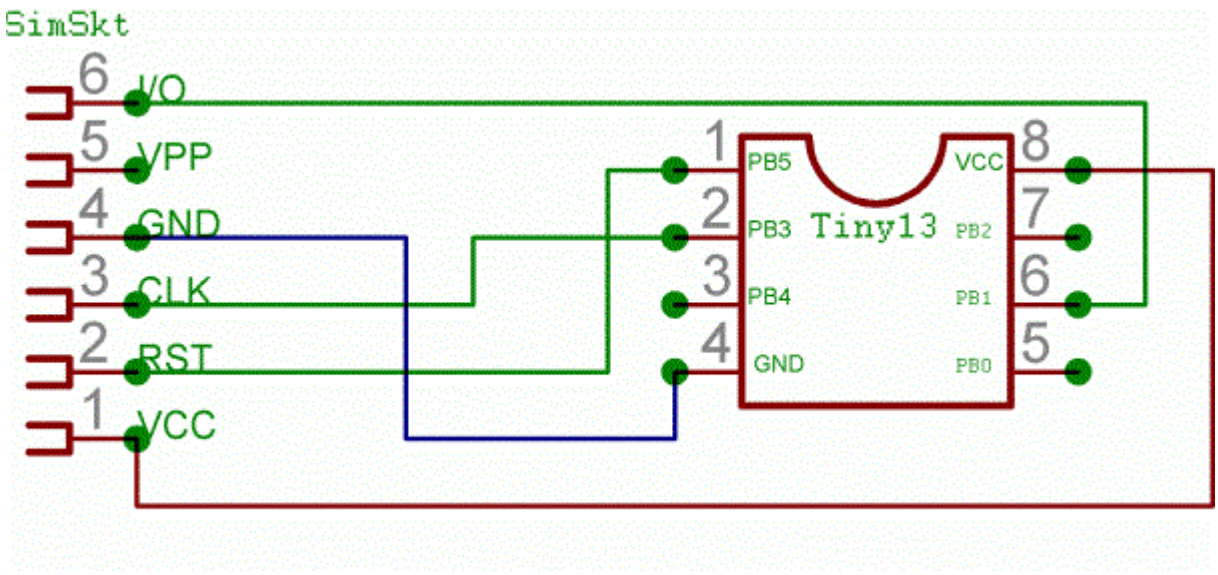
So for the range of voltages in which we're interested, viz 4 to 6 Volts, the above formula gives us a Hexadecimal value of **$0174** for 4Volts and **$03FF** for 6Volts. Since , for this application, we don't need the full 10 Bit resolution we can select the 8 Bit decimated resolution by setting **ADLAR** = 1 in the **ADMUX** I/O register. Additionally ,since we only want to transmit the voltage range 4 to 6 Volts,We subtract the digital value for 4 Volts from our computed sample and transmit the Voltage range 0 to 2 Volts as a digital representation to the Host Avr. The host uses this value to PWM the charging circuitry, so as to keep the Accu fully conditioned.

This is a simple application, written to gain some experience of programming the Tiny13.

Hope it is of interest to You.

For more Info contact Here

Schematic Simplex.asm



Schematic Accu.asm