What is the J&GBOT all about?

Translation: J&GBOT "John and Garry's box-of-tricks" - as seen at MELBPC, May 2014 Microcontroller Group meeting.

Well it started with my ArduinoMega2580R3 which I bought on eBay very cheaply.
I added a 4x4 keypad to enter data and commands and (mine has) a 16x2 parallel LCD display for interactive output. Garry has a 16x4 serial LCD display.

Next I bought an HC-SR08 module containing an AD9850 DSP signal generator on eBay. (Garry bought 2 also). They are ridiculously cheap (about 1/4 the price of the DSP chip!) and are rumored on the internet to be defective production being disposed of?

Well they work fine! –

a) The crystal oscillator gets a bit warm on some units. We have fixed this problem very simply with a 5c resistor. The oscillator now consumes much less current, runs much cooler and presumably, this has lengthened its lifespan.
b) The sinewave output tapers off at higher frequencies (presumably due to filter performance). The output is adequate and can be compensated, externally.

Most applications need only 1 HC-SRO8. (Barry's application has 2 running simultaneously)

- **Frequency Generator**

The basic functionality is to key in (typicaly 7 or 8) digits to define an output frequency and ouput it.

The keyed-in digits are displayed on the LCD.

On pressing "D" the device produces that frequency, both as a sine wave on one output and as a square wave on another output. There are a number of additional refinements provided by the software - saving, recalling, editing, etc

Current frequency range is 1Hz to about 40MHz, with sine ouput amplitude tapering off at the mid-high end. The square wave output is OK up to about 2MHz. Above about 2MHz it degenerates into little more than a distorted sine wave. External hardware fixes are planned to remedy these deficiencies.

While primarily intended as an RF generator, it is also a very useful audio frequency generator.

This current capability was demonstrated at the May Microcontroller Group meeting. Garry's software can now accommodate both types of LCD display and either 1 or 2 HC-SR08 modules.

Accuracy is controlled by the crstal oscillator - unknown, but estimated to be about 100ppm. It is expected to improve this to about 1ppm with future improvements.

- **Upgrade to manual and automatic frequency sweeping**

With no additional hardware, simply software additions, Garry now has the J&GBOT producing two types of frequency sweep mode, **Manual** and **Automatic**. It also produces a sawtooth waveform, synchronised to the sweep, for use as a CRO time base.

In both scan modes, the user enters a "start frequency", a "stop frequency" and a "step value" (all in Hz), followed by a sweep rate "step" value (in mS/step). If a "step" value of "0" is entered the sweep is manual and advances 1 step each time a key is pressed. In manual mode the current frequency is displayed on the LCD screen at each step.

- **Test Equipment Applications for Sweepers**

Automatic sweeping is very useful for adjusting tuned circuits and displaying the characteristics of filters from audio though to RF.

Manual sweeping allows precise measurement of filter, crystal, etc., frequencies. With some additional hardware , a sweeper forms a key part of a reactance tester for measuring capacitor and inductor characteristics.

[* This is where we are right now (24 May 2014) in prototype form, with/without remaining bugs.
Garry has a specific, immediate application in mind for his unit at its present state of development.
I plan to add more capability, as below and use mine as a multi-purpose test instrument *]

- **Precision Frequency Counter**

This will be the next significant capability to be designed and implemented.

It will require the hardware additions of a comparitor IC (to amplify and square up the input signal to be counted) and a precision 1 second clock source to time the count. I have a timer, gutted from a $2 Chinese crystal controlled clock mechanism. It has an estimated accuracy of about 4ppm and will set the accuracy limits of the counter.

The Arduino's counting capability is limited in speed, so an external pre-counter and use some additional input pins will be required to extend the speed range. With additional software, a counter frequency range to well beyond 30MHz is expected.

- **Temperature Control/Compensation**

The accuracy of the frequency counter can be improved by calibrataing its 1Hz timer clock against a precision 1Hz standard. We have a low cost GPS module we can use to do this. It has a 1 Hz output, accurate to within 30nS (.03 ppm)

Then there are two alternative timer clock accuracy improvement techniques avilable - **Control** or **Compensaton**.

1) **Control**-The timer clock can be controlled by attaching a temperature sensor and a heater to it and operating it at a constant temperature (say 40C?) in an insulated oven. The Arduino can measure the temperature and activate the oven in a servo loop. The clock electronics module is quite small and it is practical to build an oven around it. However, ovens have quite a high power consumption (several Watts) and long warm-up times.

2) **Compensation** - The alternative technique is to pre-calibrate the clock timer's frequency at several temperatures (from 0C up to 50C, say). From these measurents it is possible to deduce a formula to predict timer correction values at any temperature within the range.

   If the clock timer is enclosed to avoid rapid temperature changes from draughts,etc., then the Arduino can measure its temperature before and after each count to ensure that it is stable and calculate the current correction value. It can then adjust the current count value accordingly.

   With this technique, high accuracy can be obtained with little complexity and low power consumption.

The same compesation technique can also be used to significantly improve the accuracy of the frequency generator.

The total cost of all the above material is only about $50!