

SURREY RADIO CONTACT CLUB

JUNE 2007

Club Net 1.905 MHz Sunday 9.30am
Natter channel 144.215MHz Daily 8.30pm

Monthly Meetings 1st and 3rd Mondays 7.45 for 8pm

Normally at T.S. TERRA NOVA, 34 THE WALDRONS, SOUTH CROYDON CRO 4AZ

A MEETING 4TH JUNE Bar-B-Q at 113, Bute Road, Wallington QTH G4DDY

B MEETING 18TH JUNE Fix it etc (back at Terra Nova)

V.H.F Field day meeting at QTH JOHN G8IYS 11th June 8pm

From the Chairman, Pat G4FDN:

It is with great sadness and shock that I have to report the death of Mary Howells, XYL of Ray G4FFY, the club's Honorary Secretary. Mary died suddenly and unexpectedly on Thursday morning, 31st May. Ray himself has not been in the best of health for the last couple of years and I am sure that he and Mary will be in all our thoughts and prayers at this time. See www.g4ffv.co.uk

We have not been able to produce the normal format newsletter this month, for reasons most members will understand so I have included below my normal "Blog" contribution.

Chairman's Blog:

I'm sure I'm not alone in thinking last month's construction evening in building the EI9GQ PIC Frequency Counter from the October 2006 RadCom was a great success. Gareth G4XAT obviously put many man-hours in sourcing the parts, preparing the PCB's, programming the PICs and putting together the kit bags, and I think those who had the fun of building the counter are very grateful for all his efforts. At the end of the evening most people's counters were up and working. Like any construction project the benefit is not only from the practical use but the opportunity to increase one's understanding.

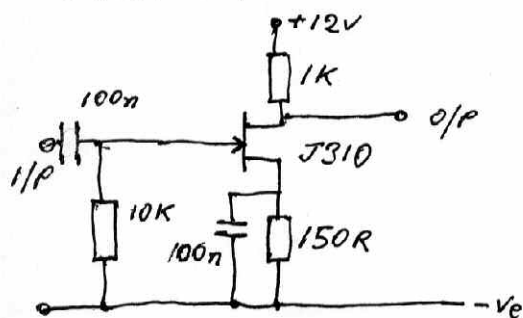
I've included below a couple of pictures of the one I built, first one starting up with the LCD in a breadboard panel and the second one when it was connected to a frequency source.



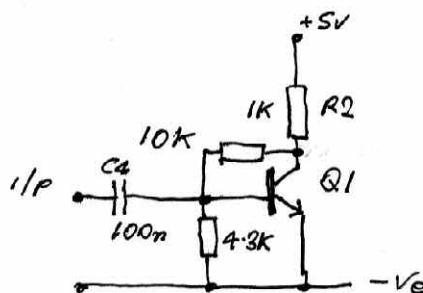
On the left of the above picture you can just see the leads I used to connect the power and input signal to the counter. These were re-cycled computer motherboard connectors and leads from the reset switch and speaker.

Like the CW PIC counter from last year, EI9GQ's design really needs an additional pre-amp stage. Also, the bias arrangement for Q1 is not really optimised and the value of R1 may need to be adjusted to get the optimum 2.5V on the base terminal. This could also be changed to a more traditional potential divider arrangement to make the bias point more predictable. Also I think the choice of a 470pF input capacitor for Q1 is too low and limits low frequency sensitivity. I think a value of 47nF or 100nF would be more appropriate. I still had the JFET input pre-amp that I built last year and this usefully increases the sensitivity of the counter to a few tens of millivolts. Though not shown in my diagram it would also be a wise precaution to protect the input of the JFET with a couple of back-to-back diodes to limit the input amplitude.

JFET PRE-AMP



MODIFIED I/P CIRCUIT FOR PIC CW FREQ COUNTER



I was quite surprised to find that in comparison to the minimal number of hardware components the complexity of some of the software subroutines was quite high. For example, the gate times for the counter are implemented in software and I assumed it would be straightforward to modify the code for different clock crystals. Not so. I reverse engineered the code for the subroutine ms100 which gives a 100ms delay to see what the underlying algorithm was that determined the constant values loaded into the count1, count2 & count3 registers and was surprised to find that the expression for the delay in terms of clock cycles was a polynomial with three unknowns. I did e-mail EI9GQ to enquire if there was a simpler method to work out the constant values. He replied below:

Hello Pat, thanks for your message and sorry about the slow response. I have been busy with the July homebrew column for the last few days.

I think you already have a good understanding of the problem.

It is quite easy to calculate values for a simple delay loop. Just figure out the time required for each iteration of the loop and multiply by the total number of iterations.

The development environment I use even has a handy tool called PiKLoops to calculate delay loop constants.

The trouble starts when you have a very complicated loop or a loop that calls a nested subroutine like the onesecond subroutine does. The onesecond routine was tuned by hand using a PIC simulator, a calculator and a lot of swearing.

The other delay loops like ms2 and ms100 are not all that critical and probably have quite significant errors in them. Most of them were lifted from a counter/stabiliser project that I built about 10 years ago.

<http://homepage.eircom.net/~ei9gq/stab.html>

I am planning to re-visit the counter project in the not too distant future. The original circuit was kept as simple as possible so that newcomers would be encouraged to have a go at building it. The revised version will have a UHF prescaler and switchable gate times. I might also change to clock frequency to 10MHz or 16.384MHz so that a surplus OCXO can be used as the clock.

VY 73, Eamon 'Ed' EI9GQ. Cork.

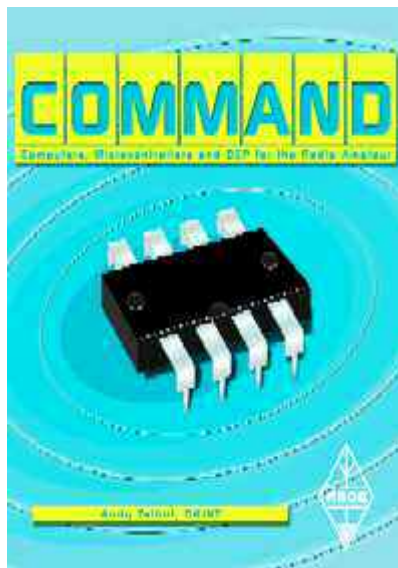
Patrick McGuinness wrote:

```
> Dear Mr Skelton,
>
> Just writing to ask a question about the code for your counter project
> from the October 2006 RadCom. I recently built this and it is working fine.
> Though I have some experience with assembly language programming from
> about twenty years ago I am a relative newcomer to PICs. I would like
> to understand better the sections of code that are dependent on the
> clock frequency and believe these are in the subroutines:
>
>         onesecond
>         ovcount
>         deci
>         ms100
>         delay
>         ms2
```

> What is not obvious to me is how the hex constant values within these
> routines have been calculated, i.e. if I were to change the clock from
> 10.24MHz to 10MHz or 4MHz how would I derive the new values? I realise
> the instruction cycle time is 4 times the reciprocal of the clock
> frequency. I have also looked at MicroChips AN592 also to see if that
> would help. I would be very grateful for any insight you can give.
> Kind regards,
>
> Pat McGuinness
>
> G4FDN (also now EI2JW)

The PiKloops utility mentioned by Eamon from <http://pikloops.tripod.com/> and runs under Linux.

To increase my understanding of PIC's in general I bought a copy of the new RSGB book PIC Basics by Cliff Powlesland, G8CQZ.



I found the first half of the book quite good but disappointingly lacking any coverage of PIC simulation. Also, I found the second half of the book covering various PIC based projects lacking in detail about design choices made. It is interesting to compare EI9GQ's frequency counter with G8CQZ design, and personally I think the former is more elegant. I have had for some time Andy Talbot, G4JNT's book, Command, also published by the RSGB. Reading this while trying to decode what the counter software was doing was a good help as I gained several insights. Subsequent to that I have obtained MicroChips (the makers of the 16F628) MPLAB integrated development environment software for PICs. I'm presently trying to get to grips with the simulation side of things in order to more easily determine how long particular subroutines take in number of clock cycles to execute. I'm not a complete beginner to assembly language programming having done a fair bit 20 to 25 years ago but I have never looked at this on PICs before. It is a pity that the source code for last years CW PIC counter has never been published as I would have liked to have looked over the innards and perhaps experimented with customizing it.

Anyway, apologies to rambling on about the frequency counter to the exclusion of everything else but I encourage club members to share their ideas, uses, modifications, etc with us in forthcoming club newsletters.

Hope to see you all at the club BBQ on Monday 4th June – now at the QTH of G4CCY & G4DDY.

Sign-Off

That's it for this month, looking forward to seeing you all on Monday 4th June at the Bar-B-Q now being held at G4CCY's/G4DDY's QTH.

73 Prim G4CCY, Maurice G4DDY, & Pat G4FDN.