PIC PULSE GENERATOR DOCUMENTATION (Version 0.21)

© T.P.Crane, 21–SEPT–1998 19:57:26.
Last revised: 17–APR–2000 16:22:26.
(1) History & Introduction

The PIC Pulse Generator is the latest in a line of Pulse Generators designed and build by the NMR group. The first unit due to Tony Kent is a standalone design based on a Z80 microprocessor with control and display via front panel keys and 7–segment LED displays.

The next Pulse Generator design used a BBC B Microcomputer to enter, edit, display and generate the pulse sequence. Counter/timers in the machine's VIA 6522 I/O chip actually produced the raw pulses which were cleaned up and made available to the outside world by a simple user–port plug–in demultiplexer box. With almost all of the work being done in software it made for a flexible easy–to–modify pulse generator. The idea for the editing display is from Brian Cowan and the software was written by the author.

Upgrading to the then new 32bit RISC processor Archimedes machines, the next pulse generator (software) used the same demultiplexer box, which was driven again by a 6522 VIA chip on the plug–in Acorn I/O card which emulated the I/O facilities of the BBC Micro on the Archimedes. Unfortunately this design suffered from delay–length 'jitter' of the order of a few μS. The jitter is due to the refresh of the computer's DRAMS, making it difficult or impossible to eliminate. While not making the system unuseable it is undesirable. As a consequence this design was never used in experimental work and prompted yet another design.

The 50MHz pulse generator was designed with very short (sub–microsecond) pulses for NMR work on solids in mind. It uses the same Archimedes data entry/display system and is also controlled in realtime by the Archimedes via the I/O card's 1MHz bus. Both pulses and delays are produced by custom 32bit, mutually triggering counter/timers inside the 50MHz pulse generator box. The logic was designed by Alan Betts. This approach eliminates the delay–jitter problems of the previous design.

There are a number of drawbacks to these computer based pulse generators which have prompted the design of the PIC pulse generator. Some of these are;

- Locks up a expensive computer for a relatively trivial job which could easily be doing something more befitting its capabilities.
- Computers and their monitors are notorious sources of RF noise which finds its way into spectrometers etc. to which the pulse generator is attached. The noise stems from crystal clocks inside the computer which together with the Fourier components of their division frequencies (production of which are an inherent feature of the busses of microprocessors) produce a broad frequency swathe of radio frequency pollution. Opto–isolation is usually only partially effective in removing the noise which is often highly frequency stable making it difficult to remove by signal averaging. This is because it can get into spectrometers via a variety of routes; Busses inside and

outside the computer/monitor can act as antennas, radiating the signals. Being high frequencies they can capacitatively couple across opto–isolators. They can be induced in the braid of coax cables, propagate via earth loops etc.

The PIC pulse generator is based around 6+ PIC16C84 or PIC16F84 microcontroller chips and a 2KByte EEPROM memory chip to store pulse sequences. Each PIC16C/F84 chip contains only 1Kword EEPROM program memory, 36/68 bytes of RAM, and 64 bytes of non–volatile data EEPROM – hence the need for the multiple chips across which the functionality of the design is split. PIC chips use the Harvard Architecture for security reasons and hence all their busses (to memory, I/O ports etc.) are internal to the chip which should eliminate the bus–radiating problems of computers and microprocessor designs using the traditional Von Neumann architecture. The instrument's display uses a 4×16 digit LCD module for data entry/display plus a few LEDs. Again this should be much less noisy than a computer monitor. The design attempts to reproduce the data entry & editing display of the previous computer based ones by using the LCD as a 'window' which scrolls up and down a 'virtual' editor display.

New Features in PIC Pulse Generator Version 0.21:

- The slave boards pulse sequences are now re-synchronised to the master board at the end of the recycle period rather than only once at the start of pulse generation. This removes the undesirable feature of version 0.20 in which the total pulse sequence length of the slaves had to be exactly the same as the master to prevent the slaves getting progressively out of sync with the master sequence every time the sequence recycled.
- For finite recycle count pulse sequences the pulse generator now beeps when it has executed the programmed number of recycles. This alerts the user that pulse generation has ceased.

(2) User Guide

(A) Initial Data Entry.

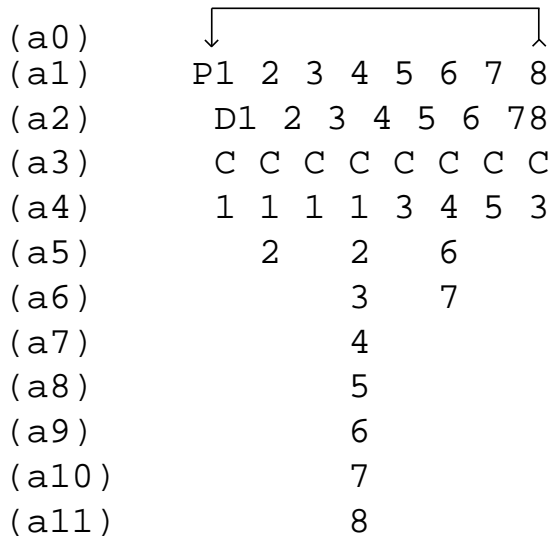This is the same as previous designs. The required input is;

- Number of Pulses in the sequence
- Pulse to recycle to
- Number of times to recycle (Enter 0 for infinity)
- The *Pn/Dn* labels and channel specifiers for the *N* pulses specified above. The *Pn/Dn* labels can be anything in the range P1–P8 & D1–D8. The channel specifiers list the channel(s) each pulse will appear on. They are lists of between 1 and 8 digits in the range 1–8 in any order. Egs. "5" for pulse#2 means the second pulse in the sequence will emerge on channel 5 only. "6254" for pulse#7 would mean the 7th pulse emerges on channels 2,4,5 & 6.
- The values and units of the pulses and delays. Values are in the range 1–16383 for pulses and x–16383 for delays, where the 'x'

depends on the position of the delay in the sequence. See specifications section below for more details. Units are one of µS, mS, Seconds or Minutes. Values are typed first and entered with the <Enter> key. Units are entered by hitting <Enter> over the the required symbol in the "µmSM" prompt.

The pulse sequence is then complete and control passes to the editor menu.

(B) Pulse Sequence Editing.

As mentioned above the pulse sequence editor is accessed by scrolling the LCD window around the virtual display using the 4 cursor keys and modifying the sequence using other key−strokes. Here is an annotated example pulse sequence from the virtual display.

```
(a0)        ┌──────────────────┐
            ↓                  ↳
(a1)        P1  2  3  4  5  6  7  8
(a2)         D1  2  3  4  5  6  78
(a3)          C  C  C  C  C  C  C  C
(a4)          1  1  1  1  3  4  5  3
(a5)             2     2     6
(a6)                   3     7
(a7)                   4
(a8)                   5
(a9)                   6
(a10)                  7
(a11)                  8


(b0)        P1=1µS
(b1)        P2=12µS
(b2)        P3=123µS
(b3)        P4=1234µS
(b4)        P5=12345µS
(b5)        P6=140mS
(b6)        P7=100mS
(b7)        P8=2089mS


(c0)        D1=100µS
(c1)        D2=12345µS
(c2)        D3=1mS
(c3)        D4=10mS
(c4)        D5=100mS
(c5)        D6=1000mS
(c6)        D7=1Sec
(c7)        D8=10Min


(d0)        Recycle# 65123
```

The editing options for each line/group of lines are given below.

(a0)   Both the number of pulses in the sequence and the recycle–to position can be altered by clicking the <Enter> key over the arrow–head or arrow–tail moving left or right to the desired new position and clicking <Enter> and second time. Naturally invalid choices such as recycle-to pulse > number–of–pulses are rejected.

(a1–a2)   *Pn* and *Dn* pulse/delay labels are modified by pressing a number key over the pulse/delay # to be modified. The number pressed defines the new *Pn*/*Dn* label.

(a3)   Pressing a number key over the '*C*' for a particular pulse will add that channel to the list of channels the pulse will emerge from. That list is displayed vertically over lines a4–a11.

(a4–a11)   Pressing the <Delete> key over a channel specifier for a particular pulse will remove that channel from the list of channels the pulse will emerge from.

(b0–b7)   Both the *Pn* pulse label values and units can be modified here.

                 (a)   To modify the value – overtype it with the new value and hit <u><Enter> over any part of the number</u> to enter it. Hit <Delete> to discard any changes made. Note: to change for example, $1000 \rightarrow 100$, the string "1000" can be overtyped with "0100".

                 (b)   To modify the units – Hit <Enter> over any part of the units string, eg. over the 'S', 'e' or 'c' of "Sec". The units chooser string "µmSM" then appears. Finally, hit <Enter> over a symbol for the desired new units.

(c0–c7)   Editing the *Dn* delay labels is the same as for *Pn* pulse labels. There are some additional issues here which relate to µS units delays. If too small a value is entered for a particular delay pointed to by a label an error message will be printed. The message will refer to a delay for which the value was too small. Note: This does not necessarily imply that delay was the only one for which the value was too small. It does not even necessarily mean that delay had the longest minimum acceptable value of the delays referred to by that label. After the error message the delay value will be unchanged and the user must leave it unchanged or choose a larger value.

         A similar situation can occur when *Dn* units are modified. Eg. if a particular *Dn* was 5mS and the user modified the units to µS an error will occur because the pulse generator cannot generate a delay as short as 5µS. The units will remain unchanged and user must first change the *Dn* value to something acceptably larger and then change the units to µS. Note: Under certain circumstances *Dn* label values with µS units can alter 'spontaneously' by a few 10s of µS when edits are made in lines a0–a2. This is an unfortunate side effect of the way pulse sequence

information is stored in the PIC chips memory together with the very limited amount of memory available in these chips. A future revision of the software may modify this behaviour. In the mean time the user should check *Dn* label values after any modifications in lines a0–a2.

There are some additional issues concerning the recycle delay in slave sequences the user should be aware of: The slave board recycle period synchronisation modification in vers. 0.21 makes the slave board recycle delay value entered by the user effectively redundant since it is now controlled by the master pulse generator – that is, the slave boards pulse sequences now recycle when the master boards signals them to. For compatibility with the filesystem and master sequences, the user still must enter a value for the slave board recycle delay. <u>It is the responsibility of the user to enter a suitably short value such than the total slave board pulse sequence length is shorter than that of the total master pulse sequence</u>. Note(1): If this condition is not fulfilled, the result should be that the slave board sequence will recycle once for every-other master pulse sequence recycle. Note(2): During alpha testing of the version 0.21 software, slave recycle values of 100μS or less have occasionally resulted in system hangs. A value of 150μS for the slave recycle period is currently recommended. Please advise the author of system hangs or any other issues related to slave recycle delays.

(d0)   The Recycle value gives the number of times the repeat part of the pulse sequence encompassed by line a0 will be repeated. As with *Pn*/*Dn* label values the number is typed with numeric keys and entered with the <Enter> key. Specifying 0 produces an infinite recycle count.

(C) Accessing the pulse sequence EEPROM based file-system.

The file-system is accessed via the <Menu> key and may be entered at the initial `"Nos Pulses ? "` prompt and at most places in the Editor Menu described above. The initial menu appears as follows;

```
Filesys Slv Brds
```
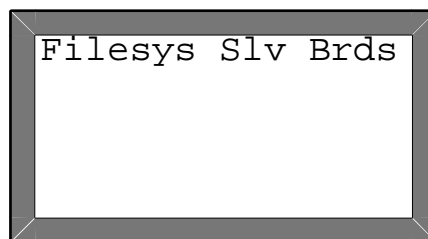
Scrolling left with the ← cursor key and hitting <Enter> over any part of the string `"Filesys"` will access the file-system menu which appears as follows, eg;

```
┌─────────────────────────┐
│  LD SV RN DL EXIT        │
│                         │
│                         │
│  12   Tom's T1          │
└─────────────────────────┘
```

From here;

> • The ← & → cursor keys can be used to chose between the options displayed on the top row (respectively LOAD, SAVE, RENAME, DELETE the current pulse sequence and EXIT the file-system menu.
> • The ↑ & ↓ cursor keys scroll through the file-system's catalogue of pulse sequences. The example above shows the 12th pulse sequence whose name is "Tom's T1" is the currently selected one. In this version of the pulse generator 26 pulse sequences can be stored by the file-system.

• Loading a pulse sequence – Press <Enter> while the cursor is over the 'LD'. If successful the system responds with 'Done.'

• Saving a pulse sequence – Press <Enter> while the cursor is over the 'SV'. The menu then displays, eg.;

```
┌─────────────────────────┐
│  LD SV RN DL EXIT        │
│                         │
│  New?                   │
│  12   Tom's T1          │
└─────────────────────────┘
```

The pulse sequence can then be saved with the same name (Tom's T1) or a new name can be entered. Hit <Enter> to save under the same name. Hit ↑ or ↓ cursor keys to enter a new name. The menu then responds;
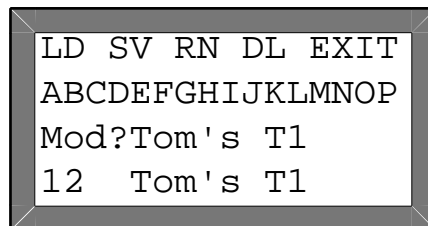
```
┌─────────────────────────┐
│  LD SV RN DL EXIT        │
│  ABCDEFGHIJKLMNOP        │
│  New?                   │
│  12   Tom's T1          │
└─────────────────────────┘
```

The new filename is constructed by choosing characters from line#2 and inserting them into line#3. The ↑ & ↓ keys toggle the cursor between lines 2 & 3. The ← & → keys select which character to choose from line#2 and where to place it in the filename being constructed in line#3. Hitting <Enter> over a character in line#2 selects that character and hitting <Enter> over a position in line#3 inserts the

selected character at that position. Hitting <Enter> over line#3 a second time completes the operation and saves the pulse sequence. Any‡ of the 256 characters generated by the display can be used in a filename and are accessed by using the ← and → keys to scroll line#2. Note: The display generates many special symbol characters as well as a variety of accented characters. Some example filenames/ pulse sequences have been pre-inserted into the EEPROM.

‡ There is one exception. The undefined character which occurs one place to the left of the first instance of the 'μ' may not be used.

- Renaming a pulse sequence – Press <Enter> while the cursor is over the 'RN'. The menu then displays, eg.;

```
LD SV RN DL EXIT
ABCDEFGHIJKLMNOP
Mod?Tom's T1
12  Tom's T1
```

The procedure for modifying the filename displayed in line#3 with characters selected from line#2 is the same as described above for saving a pulse sequence with a new name.
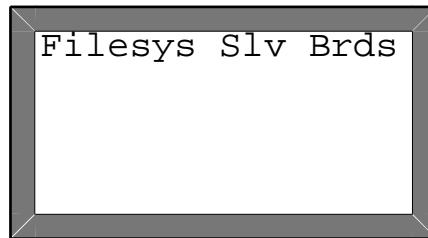
- Deleting a pulse sequence – Press <Enter> while the cursor is over the 'DL'. The menu then responds with "Sure?". Hit <Enter> to confirm the delete operation or hit any other key to back out.

- Exiting the file-system menu – Press <Enter> while the cursor is over the 'EXIT'. If a pulse sequence was loaded from the EEPROM or a pulse sequence was already present before the file-system menu was invoked, control passes to the editor menu, otherwise control passes to the initial data entry procedure for a pulse sequence to be entered.

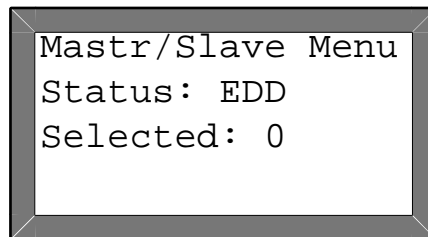(D) Operating the Slave pulse generator boards.
    The Slave pulse generator boards are accessed in exactly the same way as the Master pulse generator. Two new concepts and a new menu have been added to control the Slave boards. The two concepts are;
(1) The notion of a 'currently selected pulse generator board'
(2) The Enable/Disable status of a Slave pulse generator board.

Both entities can be displayed and altered via the Slave Board's menu which is accessed from the main menu;

```
┌─────────────────────┐
│ Filesys Slv Brds    │
│                     │
│                     │
│                     │
│                     │
└─────────────────────┘
```

by hitting <Enter> over any part of the string `"Slv Brds"`. An example Slave boards menu is;

```
┌─────────────────────┐
│ Mastr/Slave Menu    │
│ Status: EDD         │
│ Selected: 0         │
│                     │
│                     │
└─────────────────────┘
```

The Enable/Disable status of the 3 slave boards is shown on line 2 and the currently selected pulse generator board is shown on line 3. In the example slave board#1 is enabled, slave boards#2 & #3 are disabled and the currently selected pulse generator board is board#0 – the master pulse generator board. The currently selected pulse generator board can be changed by hitting a numeric key in the range [0–3], 0 being the master pulse generator board and 1–3 being the 3 slave boards. The enable/disable status of the 3 slave boards can be changed by making that slave board the currently selected board (ie. hit a numeric key 1–3) and then using any of the cursor keys to toggle its enable/disable status. To leave the slave boards menu, hit <Enter> which returns the system to the main pulse sequence editing menu.

From this point hitting the <Start/Stop> key will cause the currently selected pulse generator board to be loaded with the pulse sequence displayed in the editor menu and pulse generation (on the master pulse generator board plus any enabled slave boards) to be started.

Alternatively, pressing the <Load> pulse sequence key will only result in the currently selected pulse generator board being loaded with the pulse sequence displayed in the editor menu. The <Load> pulse sequence key permits each of the pulse generator boards to be preloaded with what ever pulse sequence is required <u>before</u> starting pulse generation. This is the way a parallel pulse sequence is built up.

Note (1) As far as the editor menu and file-system is concerned, each pulse sequence loaded into each pulse generator board is a separate entity which must be saved as an entry in the file-system if it will need to be edited or reused in the future or on another pulse generator board.

Note (2) The master pulse generator board is always enabled and therefore <u>must</u> have a pulse sequence loaded into it for pulse generation to work. If the system is booted, pulse sequences only loaded into the slave board(s) and the <Start/Stop> key is pressed then pulse generation will not start. Instead, the error message `"MPS?"` (ie.

Master Pulse Sequence?) will be displayed and the pulse sequence editor menu will be reentered.

(E) Stopping/Starting Pulse Generation.

The <Start/Stop> key starts pulse generation. It will continue until the recycle count is reached or a user hits the key again to stop pulse generation. Note: If the <Start/Stop> key is pressed during a pulse that pulse will be cut short. For this reason it is a bad idea to hold down the <Start/Stop> key in the hope of 'manually recycling' the pulse sequence. The <Start/Stop> button's LED lights when pulse generation is in progress. After pulse generation has finished the editor display is reentered.

(F) Other Features.

- Pressing the <Escape> key software resets the pulse generator and restarts it at the initial pulse sequence entry point.
- The pulse generator has an external clock input facility. Selecting this will affect all timing within the system. This is because all timing within the unit is derived from the clock. Internally a 4MHz clock is used. If for example an 8MHz external clock is applied, all pulses and delays produced will be ½ the displayed values. A front panel LED indicates whether internal or external clock is in operation.
- The 'Bleep disable' key can be used to disable the bleeper which produces the 'keyclick' etc. This a latching key and its LED reflects the enable/disable status of the bleeper.
- The system has a hardware reset button on the back–panel. This can be used to regain control of the system in the event it hangs.
- In contrast with the previous microcomputer based pulse generators the PIC system does not use a relatively slow BASIC interpreter to produce the recycle delay. Consequently minimum recycle delays are only marginally longer than those of other delays. Be aware therefore that if you erroneously specify a recycle delay of say 100μS it will happily generate it, burning out the transmitter/frying your sample etc. You have been warned!

(3) Specifications (individual pulse board):

Number of pulses generated:	$1 \leqslant n \leqslant 8$.
Number of possible recycle–to positions:	$1 \leqslant m \leqslant 8$, $m \leqslant n$.
Number of independently specifiable pulse lengths:	1–8.
Number of independently specifiable delay lengths:	1–8.
Number of channels from which any pulse may emerge:	1–8.
Minimum & maximum pulse lengths (4MHz clock):	1–16383μS, 1μS steps
	1–16383mS, 1mS steps
	1–16383Sec, 1Sec steps
	1–16383Min, 1Min steps

Minimum & maximum delay lengths (4MHz clock):	x–16383μS, 1μS steps
	1–16383mS, 1mS steps
	1–16383Sec, 1Sec steps
	1–16383Min, 1Min steps
	x=60μS for Delays#1–#2
	x=94μS for Delay#3
	x=109μS for Delays#4–8
	x=x+32μS–y for a recycle delay.
	if $n \geqslant 3$ and $m \leqslant 3$ then y=34μS otherwise y=0.
Minimum & maximum recycle count values:	1–65535, steps of 1 or unlimited.
Minimum & maximum external clock frequencies:	DC–4MHz guaranteed with current 4MHz rated PIC chips. Frequencies 10MHz–16MHz will probably work too. 10MHz rated PIC chips may be needed for frequencies >10MHz. Best to test external clock frequencies >4MHz before using on live experiment.

(4) Bugs

(a) The most complicated section of the software deals with scrolling of the LCD 'window' within the virtual editor display. It is difficult to eliminate all bugs from such code. Occasionally after certain edits the LCD is known to show erroneous entries such as something like;

```
P1=1μS
P1=1μS
P2=123μS
P3=1234μS
```

where the P1 pulse label is incorrectly written twice. This a benign bug and can be cleared by scrolling up/down the display causing the bad section to be repainted. If this is a problem, please make a note of the exact circumstances under which it occurs to aid debugging.

(b) For pulse sequences containing no *long*[†] pulse or delay values, pulse generation cannot be stopped with the <Start/Stop> key – Only a hard–reset will halt pulse generation. Note: Only one pulse or delay (eg. the last/repetition-time delay) in the pulse

sequence needs to be *long* to avoid this bug,

† 'Long' means > 32µS for pulses and typicallyu 150µS for delays.

(c) Due to the fact that PICs internally divide their clock input frequency by a factor of 4, it is possible that there will be small differences in time between the start-of-pulse-generation-time on the master and any of the three slave boards. These differences will always be less than 1 machine cycle (ie. <1µS with the internal 4MHz clock) and will remain constant until the pulse generator is Reset or switched off/on again – that is, it will not result in pulse jitter. More recent versions of the PIC pulse generator have extra H/W to eliminate this problem. If this is a problem on older (eg. 19" rack mounted units) please consult Alan Betts regarding possible modification.

(5) Possible Future Enhancements

- A computer interface to permit a computer to drive the pulse generator. Specifications for this are currently undecided. Possibilities are GPIB/IEEE–488, RS232, I2C, 8–bit parallel port, Ethernet etc.
- If RF noise from the system is a problem it may be possible to use the PIC's SLEEP instruction to put them into a dormant state when pulses are not being generated or the system accessed by a user etc.