

# A Software based controller for RF Tube PA's

Dave Powis DL/G4HUP

## Abstract

This paper discusses the suitability of microcontrollers for implementing the functions required in monitoring and controlling circuits for RF tube PA's, and presents a comprehensive design based on the Microchip 16F84 PIC . Although the paper is focussed on the software and controller aspects of the design, some discussion of the necessary hardware interfacing will also be presented. The hardware design for which this controller was developed was done by Guenter Koellner, DL4MEA, and a full description of his design is not within the scope of this presentation - enough detail will be discussed however, to enable other implementations of a complete PA control system based around this PIC development to be designed.

## Introduction

There have been many articles and presentations about PA controllers - some good references are to be found in every amateur construction manual, eg RSGB, ARRL, etc. Particularly good examples are those of G3SEK [1] and G4FRX [2]. The purpose of introducing a microcontroller into the control process is not to provide extra functionality, but rather to simplify some of the logic and interlocking - certainly, it is relatively easy to introduce quite complex interlock relationships, giving better flexibility, protection and therefore reliability of the PA overall.

Following on from my earlier experimentation and self education in microcontroller development through the programming of PIC beacon keyers [3], I was looking for another suitable subject to further my development experience. A discussion with Guenter about PA's and control circuitry in general lead us to the possibility of putting a substantial proportion of the necessary functionality into a microcontroller, leaving only the signal conditioning and interfacing to be built around it. Like all projects, the scope changed during the early stages - initially the specification was for a controller that could be used for tube and transistor PA's, with full sequencing for all changeover functions included. As the investigation proceeded, this was reduced to a final specification which was for tube PA's only, and used sequencing external to the PIC. These changes were partly due to rationalising the functionality required, especially on the monitoring and supervisory side, and also to the practical limitation of matching the functionality with the I/O capability of a suitable processor.

## The Requirements and Assumptions

The final agreed requirements for the controller board was that it:

- would be usable with triode and tetrode tubes
- would have the warm up delay variable for different tubes
- would operate in normal and bypass modes (ie Tx/Rx with PA on, and Tx/Rx with PA off), with Bypass mode automatically selected while the heater warm up delay is taking place.
- must provide sequencing for the PA relays and preamp changeover
- would have both  $I_g$  &  $I_k$  overcurrent monitoring with warm and cold restarts, depending on the fault condition
- would have blower control included - continuous, or Tx only with temp sensing.
- would have a user definable monitoring input, eg for air flow detection, as well as G1 and G2 monitoring
- would have interlocked operation of heater, and EHT

Some assumptions were also made:

- the warm-up delay need not be fully variable - 1, 2, 4 and 5 minutes would give delays to cope with almost any tube likely to be found in amateur service, without having to wait too long!
- all input signals would be active low - logic 0 = condition on, since in many cases a single transistor could then be used for signal conditioning
- all output signals would be active high - logic 1 = condition on for driving interface circuits.
- bypass operation would automatically be resumed if the controller detected  $I_k$  overcurrent and went into cold restart

## Operation Modes

Two modes of operation have been mentioned above - Normal and Bypass. These are defined below:

## **Bypass Mode**

When the PA is off, or while the heater delay is maturing, the presence of ground (0v) on the PTT line will cause the Preamp Relay output to go into Tx mode. It will not allow the PA itself to change into TX, and will not allow the drive signal to be fed into the amplifier - thus the driver stage will be connected directly to the antenna on Tx, but any preamps in use will be fully functional on receive

## **Normal Mode**

Once the heater delay has matured, then the presence of ground (0v) on the PTT line will cause a full change over to Tx, with the drive being fed to the PA, and the PA connected directly to the antenna on Tx. On Rx the antenna will be connected directly to the driver, or via the preamp if it is switched in.

Additionally, in each of these operation modes, during Rx, the Rx Preamp may be either on or off.

## **The Specification**

Working from the requirements a set of specifications was drawn up for each requirement, to determine precisely what action was necessary for each input and each combination of inputs, where appropriate. These specifications were applied at the board level, rather than the microcontroller itself, but the program specification for the chip must be derived from the overall specification. Thus, from the requirements list above, the complete controller must:

- have inputs for monitoring heater, Vg1 and Vg2 voltages, plus one 'unassigned' input for the user
- have an input for monitoring the EHT voltage (to ensure that no drive can be applied with EHT present)
- have an input for monitoring the blower mode (switch input) and an input for monitoring a temperature sensor
- have inputs for monitoring grid and cathode overcurrent conditions
- have an input for the Preamp On switch
- have outputs for controlling the Preamp and PA relays
- have outputs for controlling the EHT
- have an output for controlling the blower.

## **But why use a Microcontroller?**

All of the foregoing are standard requirements for tube PA controllers - there is nothing there that makes it essential to use a microcontroller. There are some advantages to using one - there are also some disadvantages to be considered:

**Advantages** (in addition to the basic challenge of implementing it!)

- replaces all the hardware logic (well - most of it!)
- relatively easy to implement complex relationships and interlocks
- easy to change the logic - no re-wiring or circuit changes needed

## **Disadvantages**

- all inputs and outputs have to be isolated, to protect the device
- signal conditioning is required for all inputs and outputs - ie all inputs must be between 0 and +5V, regardless of the parameter being monitored, and all outputs must be converted to the actual drive levels required
- a separate power supply is needed (a supply derived from a common transformer is OK, provided it has its own independent rectifier and regulator circuit)
- since the controller has a clock, it is a potential source of EMC problems, and must be screened

Of the disadvantages, only the last two are really significant - because of the voltages around a tube PA, some signal conditioning will be needed even for a hardware controller, and in fact the use of opto-isolators helps in getting to the right voltages, as well as providing the necessary isolation.

The use of a separate power supply is recommended to support the isolation mentioned above - after going to great lengths to isolate all the inputs and outputs, failure to provide a separate supply would compromise this. Ideally a separate winding on the transformer should be used, but as a minimum requirement a rectifier, smoothing and regulator circuit dedicated to the microcontroller should be used.

Screening and filtering/decoupling of all connections is good practice anyway, since with high RF fields floating about from the PA itself, even a hardware controller can have problems - in this case, however, the microcontroller clock is running continuously, so it could cause problems during reception if not properly screened.

## Controlling the Relays

Given the definitions of the operating modes above, and the requirements for when these modes will be used, the following diagrams explain the status of the relays for each mode.

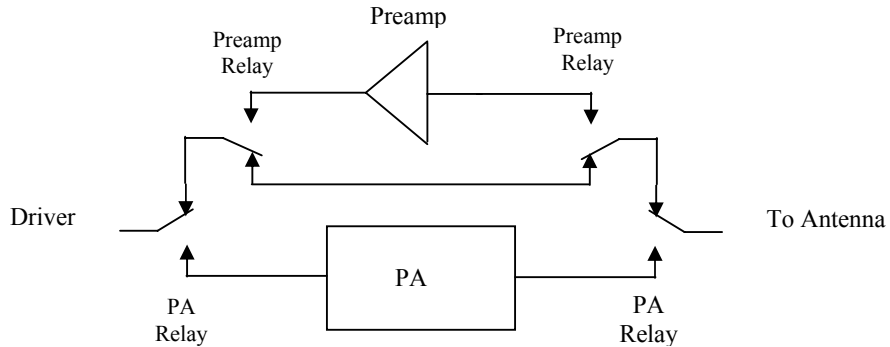


Fig 1 Switch positions for Bypass Mode Tx, and Rx (either mode) with preamp off

Fig 1 shows the default position for all the relays - this gives full bypass of the PA (and Preamp) and shows how the relays must be connected in their 'off' state. This ensures that in the event of any supply failures, there is still a connection from the antenna to the driver. This is also the state that the controller must put the relays in for the Bypass Tx mode, and also for Rx with no preamp.

Fig 2, below, shows the relay positions for Normal mode Tx - ie when the PA is operating. In this case, the position of the Preamp relays is irrelevant - they have been shown in the 'preamp on' position.

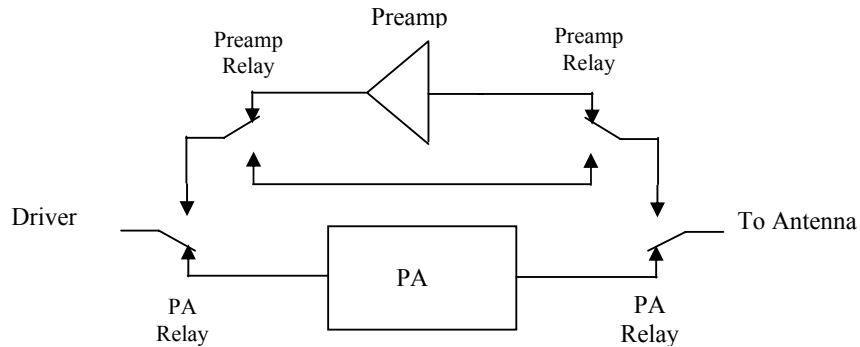


Fig 2 Switch positions for Normal Mode Tx

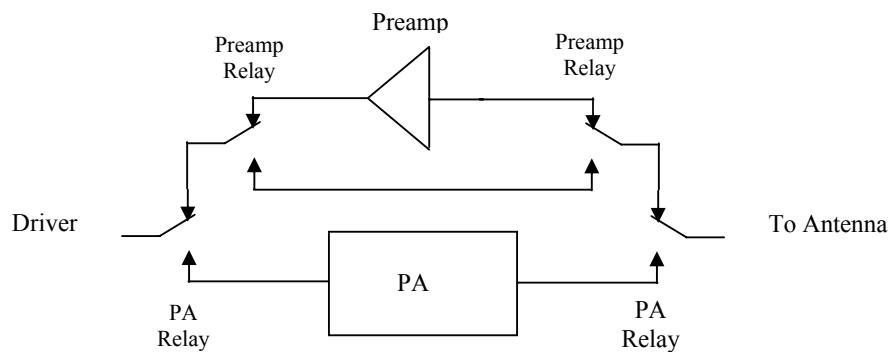


Fig 3 Switch positions for either Mode Rx, Preamp on

Finally, Fig 3 shows the positions for Rx mode with Preamp on.

Two significant conclusions which will have an impact on the control logic can be drawn from these diagrams:

- Although there are two operating modes, and two variants of Rx state, there are three different states of the relays to be considered, covering all possible valid situations.
- The Preamp relay only needs to be switched when the preamp is on (obvious, of course!) - but this means that the microcontroller does not actually need to know about this. If the microcontroller always switches the Preamp relay on a Tx/Rx change, and the output is switched through the 'Preamp On/Off' switch, the correct condition will be applied.

Table 1, below, summarises the switching needs of the controller

State	PA Relay	Preamp Relay
1 - Rx	off	Depends on Preamp switch
2 - Bypass Tx	off	off
3 - Normal Tx	on	off

Table 1 Relay switching for Tx/Rx

## Choosing the Chip

Based on an analysis of the requirements, the prime requirements for the microcontroller itself are:

- 8 input ports minimum
- 4 output ports minimum
- operation at 10MHz

The 16F84 controller was chosen, as it has 8 input and 5 output ports, of which one input can be used as an Interrupt. It also is available in a 10MHz version, as well as the standard 4 MHz part. It has a Timer and Prescaler function built in enabling the generation of accurate timing delays, and these can also be used to trigger an Interrupt

The 16F84 is an 18 pin Flash programmed part - ie it does not have a UV erase window, but is electrically programmable and erasable - it is a direct equivalent of the 16C84 EPROM part. This feature has been found very useful during development, since the cycle time for erasing and re-programming is much shorter for the Flash version!

## Input /Output Ports

### Controller Input Pins

The following paragraphs describe the operation of the eight input pins of the microcontroller

#### Heater On Input

Monitors heater supply to PA tube. When input is high, heater is off, and PA will be used in 'bypass' state - ie operation of the PTT will set the Preamp relays to Rx or Tx as appropriate. When input is low, ie active, heater is on, and tube warm-up delay must mature before any further actions are taken. Once the delay is achieved, EHT can be applied, and the PA can be operated.

#### Grid & User Supervisory Input

Monitors presence of grid 1 voltage, externally ANDed with any other voltage the user may want to monitor - eg grid 2 for tetrodes. Once heater delay has matured, this input will be read before any further action is taken - if active (ie low) then EHT can be applied. If high, then no further action.

#### EHT Present Input

Monitors for presence of EHT supply - once heater delay has matured and G1 is present, then EHT on is checked - if present then any subsequent PTT active input will cause change over to full Normal Tx. If EHT not present, then PTT active input will cause the controller to change to Tx Bypass mode.

#### PTT Input

Active low input (ie ETT) driven by PTT line of driving transceiver.

## Blower Mode Input

Input to control Blower operation. When input is high, blower will be on whenever PA is in Tx state. When input is low, ie active, blower will be switched on when Tx is started, and will remain on during Rx until Temp input goes high.

## Temp Sense Input

Monitors PA cooling air temperature. Input is ignored unless Blower Mode Input is low. In this case, input from Temp Sense is read on switching to Rx. If Temp Sense is low, then blower remains on until Temp Sense goes high, or until Tx state is required.

## Grid Overcurrent Input

Monitors the grid current and when excessive grid current flows will switch the PA back to Rx, removing the EHT, but not resetting the timer. The PA cannot be put back into Tx mode until the PTT has been released and re-applied.

## Cathode Overcurrent Input

Monitors the cathode current of the tube, and when excessive current flows, eg a flashover, switches the PA back to Rx, removes the EHT, and resets the heater timer. Normal mode Tx cannot be resumed until the heater on delay has matured again. Whilst this is happening, the PA can be used in Bypass mode.

## **Controller Output Pins**

The following paragraphs describe the operation of the five output pins of the controller

### EHT On

When active, the EHT for the PA will be applied to the anode - external interfacing required!

### EHT Standby

Provides an antiphase output to the EHT On, to allow for control of voltage reduction in the PSU during Rx - this can be useful on some PA PSU's. Could be derived by externally inverting the EHT On output.

### Preamp Relay

Drive for the Preamp changeover relays.

### PA Relay

Drive for the PA changeover relays

### Blower

Output control for blower - either on during Tx only, or will remain on after change to Rx if temp sense shows temp still high - depends on condition of Blower Mode and Temp Sense inputs

Three internal flags are required for the controller:

- Heater Timer started - set when timer has been started
- Heater Timer matured - set when timer has completed
- Last PTT state - tracks PTT state when last examined, to enable detection of PTT state changes.

These are implemented as bits within a single register, since each flag is a simple binary function.

## **Fault Monitoring**

Aside from the removal of any of the enabling conditions for Tx mode, two specific parameters are monitored for indications of fault conditions:

- $I_g$  overcurrent input,
- $I_k$  overcurrent input

Grid overcurrent is tested on every pass through the test loop - the action of the controller on detecting excessive grid current is described in the Input Pin Description section.

Cathode overcurrent is monitored by an external interrupt, since immediate action is required - at any point in the test loop sequence, cathode overcurrent will cause the PA to have the EHT removed, and the heater delay reset, as well as all the relays being driven back into the Rx state, regardless of which other test was in progress.

## **Software Processes**

Apart from the controller initialisation, most of the processes are simple input test and branch type operations, resulting in the change of state of appropriate output pins according to the actions required. The two exceptions to this are the heater delay timer and the cathode overcurrent detection. These are both implemented as Interrupt routines. Cathode overcurrent has already been described above.

### ***Heater Delay Timer***

The heater delay timer has been implemented as an interrupt because this enables the controller to monitor the PTT input pin whilst the timer is maturing, and thus allows the Bypass mode to be used. Other methods of implementing the timer, eg delay loops, would consume all the processing capability, making Bypass operation during heater warm up impossible.

The timer uses the Prescaler and timer functions of the controller, with the Interrupt flag set so that when the timer matures, the Interrupt Service Routine (ISR) is called. The timer is started by a routine which is called the first time that Heater On is detected. This routine loads the counter and prescaler default values, and sets the parameters for the Interrupt flag. Once the timer has been started the counter decrements once for every clock pulse of the controller, regardless of what else the program is doing - so the delay is independent of whether the PA is being used in Bypass mode, and whether it is in Tx or Rx.

A nest of three variables is used to achieve the long delay required - even then, it is still necessary to use the prescaler on its maximum division ratio.

A single ISR handles both the timer and the cathode overcurrent - again the test and branch approach is used first to determine which condition has caused the interrupt, then separate routines handle the conditions and set the correct return values for subsequent operation.

When the heater delay has completed, the Heater Timer Matured flag bit is set - on subsequent passes through the main program loop, control will then pass to the  $V_g1$  monitor.

### ***I<sub>k</sub> Overcurrent Interrupt***

Once the Heater Delay timer has matured and the flag set, then the interrupt flags are re-enabled for the I<sub>k</sub> monitoring - they do not need to be set earlier, because there can be no anode current until the controller has completed the warm up cycle. Now the Interrupt will occur immediately the I<sub>k</sub> input pin changes state - regardless of what else the controller was doing, or where it was in the program loop.

The EHT is removed immediately, followed by a sequence of restoring the PA to Rx and clearing the state flags so that before EHT can be applied again, a full heater warm up period must elapse as a minimum - usually, however, the effect of an I<sub>k</sub> overcurrent event is more drastic and some intervention will be required from the operator anyway! The PA will continue to function in Bypass mode, so if the worst should happen during a contest pile up, at least you can carry on until the pile up is past before having to deal with the source of the problem.

## Program Logic

The control logic is based on simple input pin testing and branching depending on the result. Inputs are tested in the sequence:

- Heater on input
- Vg1 input
- EHT Present input
- PTT input

Fig 4 shows the MAIN program loop functions

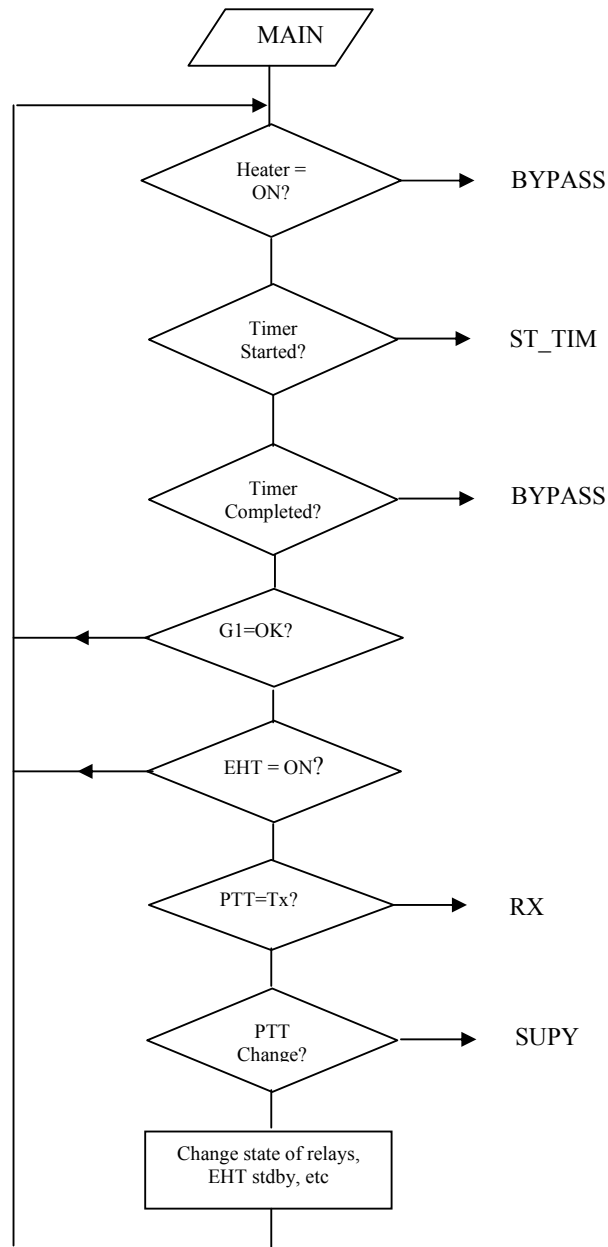


Fig 4 MAIN program loop

If the Heater is detected as active, the state of the heater delay timer is next checked - until this has matured, the software will loop via the BYPASS checking.

If any input condition is in the inactive state, then PTT is checked to test for Bypass mode Tx, before the test sequence loops.

Once the active states have been detected through the above sequence, ie the PA is in Normal mode Tx, then extra testing is performed:

- Blower mode input,
- $I_g$  overcurrent input,

These tests are made sequentially, and at every pass through the control loop these will be tested, assuming the PA remains in Tx.

On reversion to Rx mode, the Temperature Sense input will be tested to check what action should be taken with the Blower output pin

Fig 5 shows the logic for the Start Timer (ST\_TIM) routine

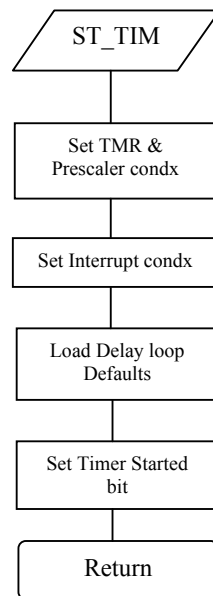


Fig 5 ST\_TIM (Start Timer) Routine

## Timing Requirements

A variable delay for the heater warm up period must be provided to cope with different types of tubes. The range of values required is between 1 minute and 5 minutes - this will cover most tubes that are in amateur use, even very large ones used by the EME community. This delay could be achieved by using input pins to define the time period to the microcontroller then selecting default pre-load values for decrementing counters - however, this will consume a disproportionate number of input pins. An alternative could be to use a controller with a built in A-D converter, and provide an analogue voltage from a potential divider - this would reduce the input to a single pin. The solution chosen reduces the pin assignment to 0 but enables any programmed controller to be used with any size of tube!

The timing has to be fixed by using a crystal - device to device variations using RC clocking are far too wide for any repeatability. . The fastest 16F84 part operates at 10MHz. At this clock rate, the controller is actually operating at a cycle rate of 2.5 MHz - a period of 400 ns. A total of  $1.5 \times 10^8$  periods are needed for a 1 minute time period. If the clock crystal is now reduced to 5 MHz, the time period will be 2 mins. Thus the delays for different tubes are handled by varying the controller clock frequency.

Table 2 shows the delay period vs the controller oscillator frequency for clock frequencies between 2 and 10 MHz.

Clock Freq (MHz)	Heater Delay (Mins)
10	1
5	2
4	2.5
2.5	4
2	5

Table 2 - Relationship between  $f_{osc}$  and Heater Delay

Since the whole program loop is only about 50 steps long, (even with the controller running at 2 MHz, the maximum time to complete the loop is 100  $\mu$ s) the time spent in the loop is very short, and switching will always be very rapid. Also, as the dangerous cathode overcurrent monitoring is by an Interrupt, then it is an immediate action, and is not related to the program loop times.

## Interfacing

As already recommended, opto isolators should be used for all input & output connections - this assists in converting the voltage levels of inputs to the correct range, and with the high current drive capability of the 16F84 output ports (25mA), there are no special requirements for the output opto couplers.

Obviously, any outputs not needed (eg EHT Standby or Preamp Relay) can just be ignored - do not connect the pin to anything. Unused inputs cannot be ignored, since every input is read at some stage during the cycle. Any unused input must be taken to logic 1 via a pull-up resistor

## Conclusions

Although the project started out as a further step in my own self education in the use of embedded microcontrollers, during this work I have become convinced of the benefit of using such a device in this application - the ease with which the logic flow could be changed (which happened at least twice during the development!), and the ability to include the full BYPASS mode operation, even during heater warm up time and in the event of  $I_k$  overcurrent in the PA. Of course, these features can be provided in a conventional controller circuit, but at a considerable increase in complexity

## Chip Programming & Software Listing

Ready programmed 16F84 chips are available from the author, or for those with programming facilities, the compiled object file can be provided. The author can be contacted at [dl4mup@qsl.net](mailto:dl4mup@qsl.net). Source files will not be made available.

## Further Developments

Although the project to produce a microcontroller based PA control board was descope to exclude the sequencing functions on the grounds of complexity, I am now developing a complete sequencer as a companion to the PA controller. This sequencer (also in a 16F84 chip) takes the Preamp and PA relay outputs from the PA controller chip, and is capable of driving normal or latching relays in either function, and also provides sequencing outputs for grid drive and ALC.

## Acknowledgements

Thanks are due to Guenter, DL4MEA, for agreeing to my publishing the software/microcontroller part of the joint project.

## References

- [1] Control & Protection for Grounded-Grid Triode Amplifiers, Ian White, G3SEK: 45 Weinheimer UKW Tagung, 2000
- [2] PA Control Circuits, John Nelson, G4FRX: The VHF/UHF DX Book, (DIR/RSGB)
- [3] PIC Beacon Keyers, Dave Powis, DL/G4HUP: 44 Weinheimer UKW Tagung, 1999