Multitasking Scriptable Autotuning PID Platform

January 10, 2016

Contents

1 Hardware

1.1 Definitons

The entirety of all physical components, resulting from this project, will be called 'device' throughout the paper.

1.2 Goals

The device is intended as a learning project for the student, but also as a open software, open-hardwre project, which anyone can create and use. Thus, the following hardware design priorities have been identified, in order of decreasing importance:

- safety - the device shall not pose a fire or electric shock hazard to the end user
- reconstructability - the device shall be composed **only** of worldwide accessible components
- longivity - the device shall remain operational for 5 years of uninterrupted service with 95% confidence
- price - the BOM for the complete device shall not exceed 100BGN
- extendability - the number of input sensors and the number of output controllers, shall be trivially configurable
- ease of assembly - it shall be possible for a person with zero hardware experience to manufacture the device
- simplicity - each component shall fufill a specific purpose, and the number of components shall be the lowest possible

1.3 Layout

Due to the requirement of extendability, the device shall consist of a number of printed circuit boards, in contrast to a single monoliotic PCB. Each PCB shall fufill a sole purpose, and any number of different modules shall be able to mate together. The following distinct roles have been identified:

- high-voltage input stage - called zero-cross detector or ZCD board from now on
- low-voltage input stage - called temperature sensor or thermometer from now on

- computational stage - called main board from now on
- high-voltage output stage - called software controlled rectifier board or SCR board from now on

The resulting design exhibits the following characteristics.

Only a single ZCD board is required, because mains waveform is invariant across the device in it's entirety. Only a single main board is required, as the selected microcontroller, although inexpensive, provides plenty of resources for numerous control loops.

In order to satisfy the requirement for simplicity, the main board is configured for a single SCR output board. However, soldering aditional connectors to the main PCB is trivial, thus acheaving extensability. The SCR output board is long-life and supports loads of up to 1kW.

The most flexible part of the system is the thermometer configuration. Due to the selected temperature sensing IC, virtually unlimited (technically up to 2^{56}) devices are supported **without any hardware changes**.

1.4 ZCD board

The ZCD board is a sensory input to the microcontroller.

Because the voltage of mains power is alternating, it is impossible to output precise amounts of power without knowing the phase of the waveform. The implementation of the ZCD board is straightforward and extremely simplified. In fact, an extensive internet search has demonstrated no other PCB has ever been designed with such a level of simplicity. In other words, **the designed PCB contains fewer elements than any known PCB for the same purpose**! This produces problems, which have deterred other designers. However, all artifats have been dealt with in software.

1.4.1 Schematic

Please refer to appendix A1 for the schematic and layout of the board.

1.4.2 Calculation

In order to protect the main board (and thus the user) from dangerous voltages, galvanic isolation is required. The standard means to this end are transformers and optocouplers. Optocouplers posess numerous advantages over transformers for our application:

- compactness
- low price
- neglegable phase shift

Furthermore, among optocouplers, the variation is considerable. We select a component with anti-parallel input LEDs, specifically designed for zero crossing - SFH620A-3. This is the most sensitive version of the IC (highest CTR), as input power is our greates concern.

Striving for minimal component count and price, the standard solution with a 10W input power resistor is dismissed. Thus, we need to work with 1/4W, E24 resistors. Due to the optocoupler's acceptable CTR, and extensive signal conditioning in software, this solution will prove to be viable!

It is worthy to note that the resistor rated voltage is of utmost importance. Because our resistors are rated to 200V peak, it would be a dangerous mistake to use a signel resistor. Therefore, the $V_{AC} = 230$V, $V_{peak} - V_{AC} * \sqrt{(2)} = 325$V is safely spread onto two identical resistors.

Let's suppose the line voltage varies from $V_{min} = 200VAC$ to $V_{max} = 250VAC$ rms.

$$P_{inputresistors} = \frac{V_{max}^2}{R_1 + R_2}$$

$$R_1 + R_2 \geq \frac{V_{max}^2}{P_{inputresistors,max}} = \frac{250^2}{0.25 + 0.25} = 125\text{k}\Omega$$

We select $R_1 = R_2 = 68\text{k}\Omega$.

$$i_{in,min} = \frac{V_{min} - 1.65}{2 * R_1 * 1.05} = \frac{198.35V}{142.8Kohm} = 1.39\text{mA}$$

The output stage:

$$i_C \geq i_{in,min} * CTR_{min} \approx 1.39mA * 0.34 \approx 0.47mA$$

$$i_{leakage} \leq 1uA$$

$$V_{IL} = 0.3Vcc = 0.3 * 5V = 1.5V$$

$$V_{IH} = 0.6Vcc = 0.3 * 5V = 3V$$

If we strive to be below 1V for logic zero:

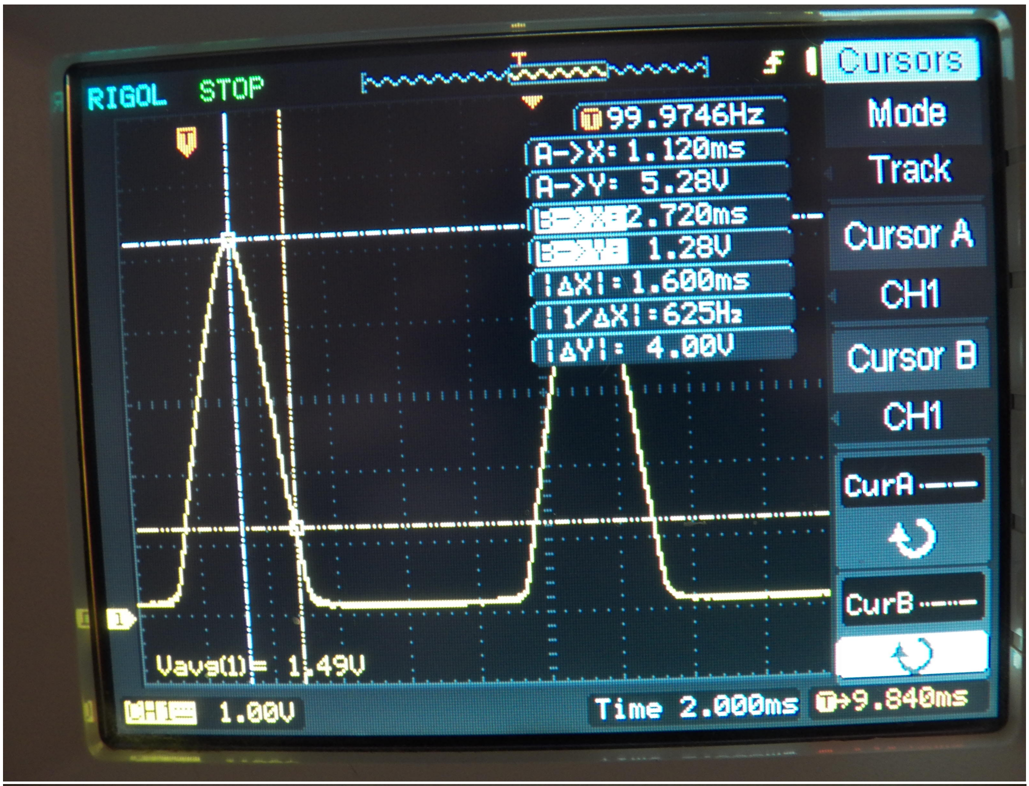
$$i_C * R_{output} = 1V$$

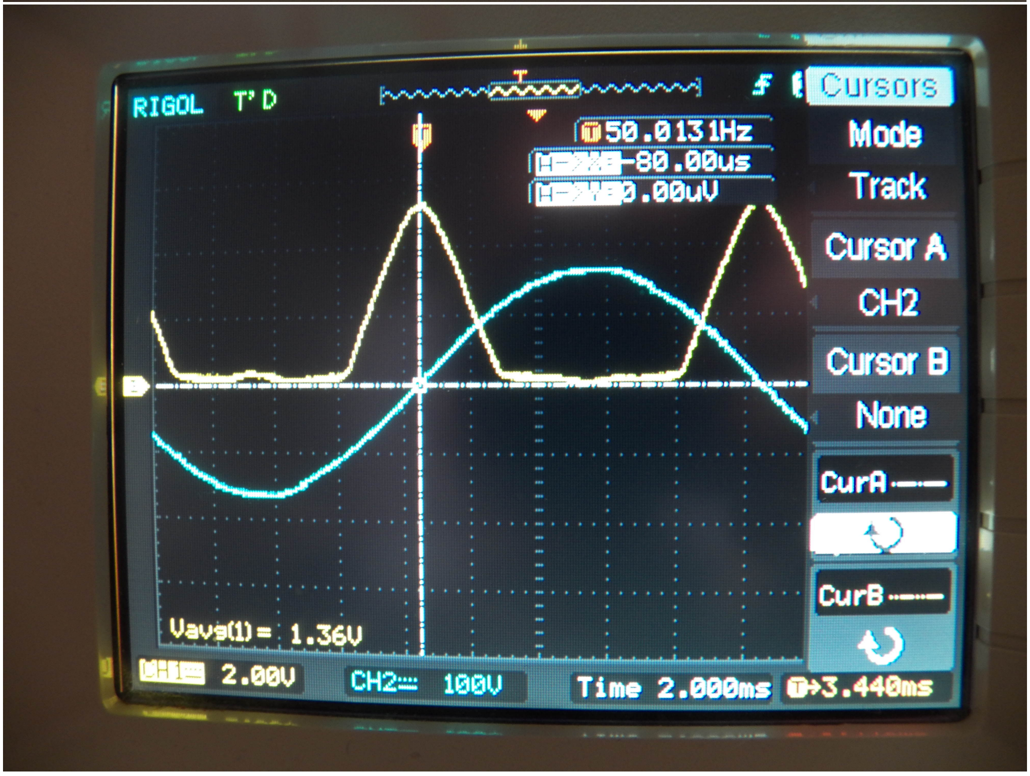
$$R_{output} = 1V/0.47mA = 2.13Kohm$$

We select $R_3 = 2.4\text{k}\Omega$.

1.4.3 Measurements

Firstly, a temperature measurement is performed. The device is allowed to run for 10 minutes. Subsequencly, each component is measured or overheating. Becuse component temperature measurement instrumentation is both expensive and dfficult to apply, the following rule of thunmb is used: *if a silicone component is too hot to keep your finger on it, it is too hot*. Althought the stated metod is vastly imprecise, it works well, because the skin pain temperature (about $60°$C) is far lower than silicone semicondutor Absolute Maximum Temperature (often $150°$C).





We observe that:

1. The pulse is very wide - about 3.2ms $\equiv$ 32% of the half-period.
2. The puslse is centered. This is great, because we can estimate the true zero crossing

in software.

1.5 Thermometer

It is impossible to examine the temperature sensing element in isolation to the heater. Therefore, in this chapter, the complete plant, or in other words the combination of heater, thermal mass and thermometer, will be examined.

As this is an educational project, the quickest possible system responce is desired. Therefore, the thermometer is directly gued to the surface of the heater. Unfortunately, even in this setup, the maximum possible heating rate is about 6°C/min and cooling is even slower.

1.5.1 Heater

Initially, a 60W incandescent light bulb was selected. As European wall power exhibits frequency of 50Hz, the highest switching frequency is 100Hz by half-periods. Astonishing to the experimenter:

- The fillament is not inert enough to integrate consecutive pulses, even if every odd half-wave is enabled.
- The human eye is unable to integrate the resulting 50Hz flicker.

As a result, looking in the controlled bulb is extremely annoying.

Consequenctly, a fish tank heater with nominal (maximum) power of 50W was selected. The observed temperature curves are equivelent sans the maddening light flicker. **The thermometer is glued to the surface to the heater for fastest response possible.**

1.5.2 Temperature sensor

The Dallas Semiconductor DS18S20 has been selected due to a variety of reasons:

- low price
- ease of interfacing to digital components
- ease of wireing
- extreme flexability of integration

This device is incedible. It can operate solely over two wires - including the ground wire. It performs digital temperature conversions, removing the need of an ADC (although our selected microcontroller features such). It can coexist on the same bus with as many as $2^{56} \equiv$ infinite number of other onewire sensors.

1.6 SCR board

1.6.1 SChematic

1.6.2 Calculation

Because we want to operate with more precision than an on-off controller, the following requirements are layed out:

- Life must exceed 3153600000 switches.
- Switching times must be in the microsecond range.
- Galvanic isolation.

In the light of this sepcification, it immediately becomes obvious that an electro-mechanical relay is not suitable. Again an optocoupler has been selected. The MOC3023 is canonical for power controll aplications. It is a triac output optocoupler, specifically designed to drive a power triac. The selected power triac is BT136 600E, providing control over applicances of up to $4\mathrm{A} * 230\mathrm{V} = 920\mathrm{W}$.

1.6.3 Measurements

1.7 Main board

The main board houses the microcontroller, provides stable power to it and contains connector blocks to the other boards.

1.7.1 Microcontroller

The selected microcontroller is atmega168 from Atmel. It provides plenty of computing power and sufficient 16KB flash program memory at extremely modest cost. It can be programmed in assembler, C or C++ with widely available and free of charge tools. The datasheet is well written and a pletora of application notes are available from Atmel. Not to mention the extensive worldwide community, readily providing help to anyone new to the subject.

1.7.2 Power filtering

Three groups of components are responsible for providing clean and steady power to the microcontroller.

Firstly, an LC filter at the power connector of the board provides filtering of the external power. This procets against insufficiently filtered power supplies and allows powering the board from a low cost wall adapter "brick". Furthermore, any EMI picked over a long power supply coard, is eliminated. Moreover, the LC filter isolates the board from the power supply, allowing the PSU to power other devicecs in the same time, free of digital switching noise.

Secondly, a combination of two capacitors, in close physical proximity to the microcontroller power leads, further conditions the power supply rail. As the microcontroller draws significant current for intervals far shorter tahn one microsecond, the PCB traces gain significant impedance. Consequently, this second group of capacitors needs to be located as close as possible to the IC. What's more, comodity electrolytic capacitors cannot operate at such high frequencies. Consequently, a ceramic capacitor is added, to handle the high frequency current pulses.

Lastly, the RESET pin of the controller is extremely sensitive to even short power line pulses (of the length of one clock cycle). Therefore, the exact RC circuit, recommended by Atmel, is used.

1.7.3 Connectors

The selected connector is NX5080-03SMR. This is a 3-pin, fixed orientation, low-current connector. The pin number is ideal for providing both power and a communication bus to connected boards. Because no terminal blocks are used, and all four connectors are wired in consistent fashion, incorrect wirein of the board is impossible.

1.7.4 Extendability

It has been paid attention to provide free PCB real estate, as well as conveniently located free processor pins. Consequently, it is trivial to add more connectors, or even a display, to the main board at a later moment of time.

2 Software

The device is a typical realtime embedded system. As such, mainframe programming tecniques are not applicable and embedded systems approach is required. Consequently, the following requirements are layed out to the program:

- Conveninet programming - during development, reprogramming the device shall be quick and easy, preferrably consisting of a single action.
- Convenient communication - both debug data and process information shall be easily accessible during device operation.
- Error tolerance - the device shall not freeze up upon an error, but should instead make best effort to keep the control loop active.
- Determinism - the programmer needs to have full control over what is executed when.
- Scriptability - the device shall be able to get reconfigured without a human at the other end id est a PC program shall be provided, which reconfigures the device in arbitrary ways.

In the oppinion of the author, all of these requirements have been fully fufilled with the current version of the software. The written software is free and open-source in it's entirety - it is protected under the MIT license.

2.1 Bootloader

2.1.1 Motivation

Several approaches were evaluated to reprogram the chip.

As the chip resides in a socket, instead of being soldered to the Main board, it could be removed from it's socket and inserted into a programmer. This approach is useful, but was discarded as extremely time-consuming.

The next possible approach is In-system programming. This constitutes soldering a header onto the Main board, and connecting it to the SPI bus of the microcontroller. This approach was also rejected for hte following reasons. The author was relucant to pay both the price in real estate on the PCB and the aditional complexity of wirein the header to the microcontroller. Not to mention using up some pins. Last but not least, this approach would significantly complicate the RESET pin nooise protection circuit.

The most complex, but in the same time best performing, approach was selected. Utilizing the fact that serial communication to a PC is needed for other reasons, the author decided to transfer new programs over the same bus. This way the device can be reprogrammed purely bu software, possibly even from a remote location.

2.1.2 Previous work

Many bootloader programs exist for the selected chip. Unfortunately, all of them are either:

- bloated with unnececery functionality, thus large in size,
- do not work out of the box and thus require manipulation of the source code to work or
- are licensed under un incompatible license than the veri permissive MIT license.

2.1.3 Implementation

The bootloader was implemented in standard-conforming C language, and written by the author in it's entirety. It is nearly impossible to create smaller in size bootloader for this device. The program code is simple and straighforward, lacking complicated preprocessor directives. Consequencly, it is in the oppinion of the author, easy to read and modify by anyone. The code is of cource publically visible and licensed in a way, that any person can freely and legally modify and redistribute it.

The bootloader instructions reside at the high end of the flash memory, while the application resides in the low end. Consequencly, the application never knows that a bootloader program was installed.

The finished product was called 'megaboot' - a bootloader for atmega devices. The source code in its entirety is provided as appendix A1.

2.1.4 Communication protocol

The XMODEM protocol has been selected. This protocol is not used in modern high-bandwidth, high-complexity networking. To the contrary. this protocol was popular when computers posessed resources comparable to the selected microcontroller. It is a simple protocol with very little overhead (channel efficiency is 97%). Furthermore, it is easy to implement in few program instruction, using up small amount of flash memory.

2.1.5 Error checking

Erorr checking is provided throughtout the program:

- CRC sums onto each received XMODEM packet
- "magic character" starts each packet
- addressing wrong (inexistient) flash pages is protected against
- receiving packets out of order is also checked agains

All the error checking code is conditionally included via the only preprocessor definition. This serves twin purpose. Not only does it clearly indicate error checking code apart from the actual program logic, but also provides the option to remove all error checking, should a person attempt to reduce the program size to the next smaller option.

2.1.6 Use

The build system passes the symbol BOOTLOAD to the application program. This symbol holds the address of the first instruction of the bootloader. The application is then free to jump to that address whenever required. In an assembler program, this would have been performed via a RJMP instruction. On the other hand, higher level languages cast the pointer to a function call address (optionally with the _Noreturn attribute) and call it.

After the bootloader has been invoked, it expects program data over the communication channel. Fortunately, many applications are available, which support the XMODEM protocol. One example is the popular and free serial terminal 'minicom'.

2.2 External components

2.2.1 Onewire library

A C library, written by Peter Dannegger, Martin Thomas and others, is being used for communication with the thermometer.

2.2.2 PID implementation

A professional implementation of a parallel pid with integral saturation, implemented by Atmel employees, is used.

2.3 Architecture

2.4 Configuration

2.5 Clock

2.6 Serial communication

2.7 Commands

2.8 Zero-cross detector

2.9 Triac cotnrol

2.10 Temperature measurement