

## Microcontroller base process emulator

Ing. Jovrea Titus Claudiu  
 jovreatitus@yahoo.com

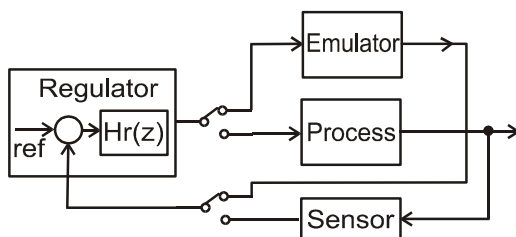
### Abstract

*This paper describes the design of a microcontroller base emulator for a conventional industrial process. The emulator is made with microcontroller and is used for testing and evaluating the performances of the industrial regulators. The parameters of the emulated process are fully customizable online and downloadable thru a serial communication from a personal computer.*

### 1. Introduction

Most of the industrial regulators that are in use are self tuning PID type. In processes that are changing their parameters in time, these types of regulators can not compensate for the errors all the time. For these processes, auto adaptive regulators can be used. In the testing period of a system with an auto adaptive regulator, the output values can be dangerous. To avoid this, the process can be replaced with an emulator which can simulate the function of the process.

The place of the emulator for testing a system is shown in figure 1.



**Figure 1** The emulator place for system test

For example, an industrial regulator for an oven can not be tested in real conditions in laboratory, because of the amount of power needed and the size of it. A method to test is: identifying the thermal process in real work conditions and then simulating it with software. After the proper regulator is found, the process can be tested in real conditions.

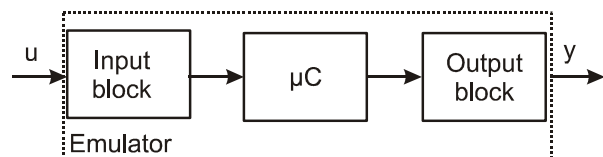
An alternative method of testing in real conditions is with an emulator instead of the real process. The advantages of its use are:

- low energy consumption; a process that require powers of tens, hundreds of KWs can be emulated with a system that requires under one watt
- the emulated process can be linear or non linear
- the testing can take place in a laboratory
- different system perturbations can be simulated
- the testing of the regulators can occur in real conditions
- the process parameters can be modified in real time trough the tests
- its not necessary to wait until the oven cools down to test it again

In some cases, the technical specification of a regulator can be faulted or can lead to confusion because some of the parameters are missing or are incorrect, methods of tuning are not specified, etc. to test the regulator in real conditions a process emulator can be used without the risk of accidents or damages.

### 2. Hardware implementation

The emulator is realized with a PIC16F874 type microcontroller, produced by Microchip.

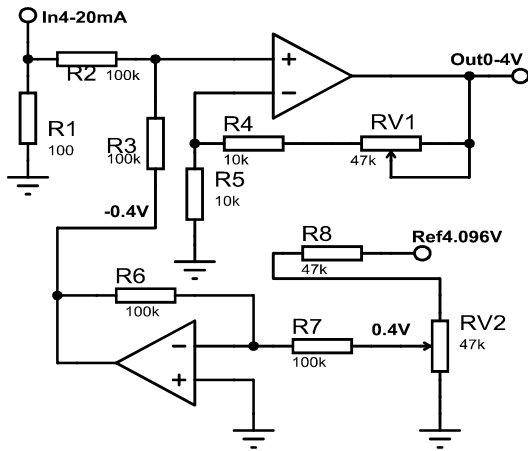


**Figure 2** Emulator block schematics

The input and output signals of the system are in the range of 4-20mA or 0-10V. The type of input and output signals can be voltage, current or the combination of them. For both inputs the signal is processed in such a way that the input voltage of the analog to digital converter (ADC) of the

**Keywords:** microcontroller base emulator, Matlab simulation

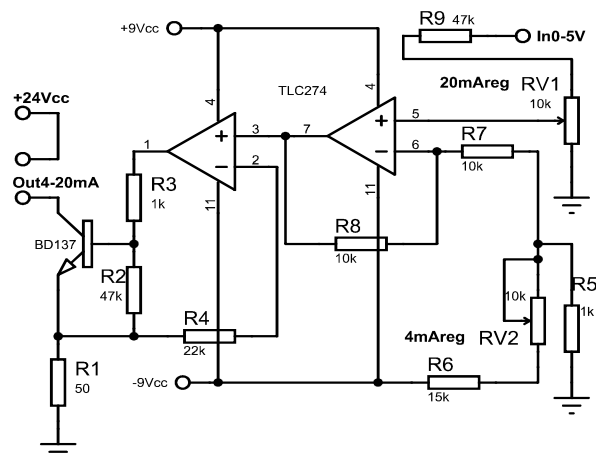
microcontroller is in the 0-4V range. The input current block for the emulator is shown in figure 3.



**Figure 3** Input current bloc schematics

The drop voltage across the R1 resistor is added with a -0,4V fixed voltage and the result is amplified by 2 with a summing amplifier. The -0,4v fixed voltage is set with RV2 variable resistor.

The output current block for the emulator is shown in figure 4.



**Figure 4** Output current bloc schematics

The current output block is a voltage controlled current source. For setting the 20mA output current, at the input we have to set a 5V voltage. The output current is adjusted with the RV1 variable resistor. For setting the 4mA output current, at the input we have to set a 0V voltage. The output current is adjusted with the RV2 variable resistor.

For the 0-10V input the signal is amplified by 0.4. Reference voltage of ADC is 4.096V and it is obtained with a voltage reference integrated circuit. The ADC of the microcontroller has a resolution of 10 bits (1024

quantization levels). The number of intervals effectively used is 1000 in the 0-4V ADC input range.

The digital to analog converter (DAC) of the system, the output, is made with pulse width modulation (PWM) block. The resolution of the PWM signal is 10 bits. The PWM signal is filtered with a first order low pass filter. The filter eliminates the 19 kHz PWM frequency to generate an output voltage proportional to the duty cycle of PWM. This is an inexpensive way to generate direct current (DC) voltage.

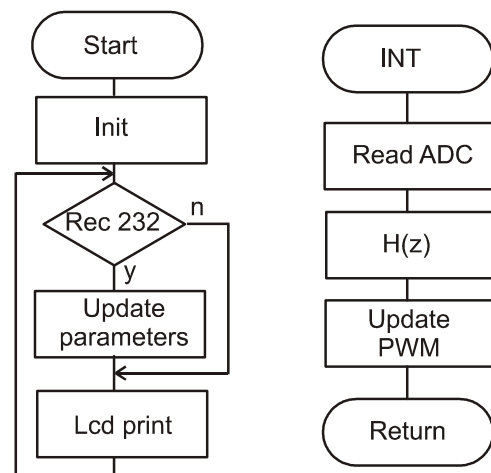
The DC voltage obtained after filtering can be processed in two ways. The 4-20mA signal is obtained with a voltage controlled current source. The 0-10V signal is obtained by amplifying the filtered signal with an operational amplifier.

Before it is used the first time, the emulator is calibrated with the variable resistors placed on the amplifier feedback loop.

### 3. Software implementation

The flow code of the program written in microcontroller is in figure5.

At a precise time interval given by the Timer1 overflow the program execute the interrupt service routine. The discrete digital numbers obtained from the ADC are fed into a buffer. The output value of the transfer function of the emulated process is computed and the duty cycle of the PWM module is updated. The buffer is also used to emulate the death time of the process.



**Figure 5** Logic diagram of the program

The program is continuously testing the serial communication port to receive the data from the

computer. The parameters of the transfer function are modified by the received data.

The transfer function model used in emulator is:

$$H(z^{-1}) = \frac{Y(z)}{U(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (1)$$

Where:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-na} \quad (2)$$

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-nb} \quad (3)$$

The transfer function in continuous domain chosen for emulation is:

$$H(s) = \frac{3.3}{0.925 \cdot s + 1} \quad (4)$$

The discrete transfer function model for a first order system used for calculates the output value is:

$$H(z) = \frac{b_1 \cdot z + b_2}{z - a} \quad (5)$$

After discretization of continuous transfer function (4) with Tustin method (bilinear approximation) with a sampling time of 0.1s the discrete function is:

$$H(z) = \frac{0.1692 \cdot z + 0.1692}{z - 0.8974} \quad (6)$$

In microcontroller the output  $y$  is calculated with:

$$y_k = b_1 \cdot u_k + b_2 \cdot u_{k-1} - a \cdot y_{k-1} \quad (7)$$

For the transfer function (6) the output  $y$  is:

$$y_k = 0.1692 u_k + 0.1692 u_{k-1} + 0.874 y_{k-1} \quad (8)$$

After reading the ADC the data is processed for the discrete function calculation. After the output  $y$  of the transfer function is found the duty cycle of the PWM signal is updated.

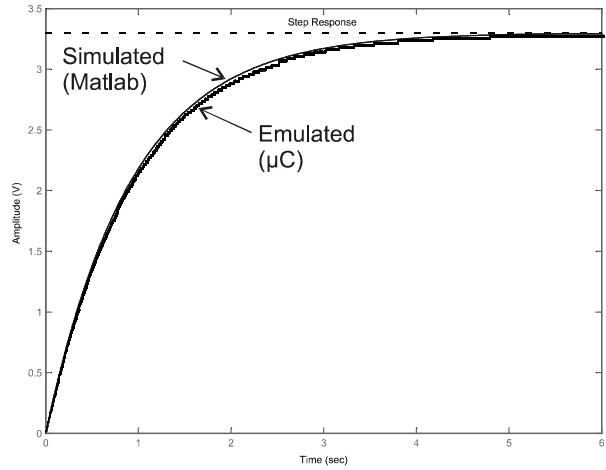
The data for the output  $y_k$  and the input  $u_k$  is stored for the next calculation of the transfer function.

For higher sampling rate the interrupt routine must be as short as possible. For example, the time spent by the interrupt routine to take a sample from ADC, to store the new data ( $y_k$  and  $u_k$ ), to calculate the output with equation (8) and update the PWM take about 750 $\mu$ s when the microcontroller is running at 20MHz. In this case a rate of 1000 sample per second can be achieved. The rest of time (250  $\mu$ s per cycle) can be used for serial communication or for printing data at the LCD. For a second order transfer function the interrupt routine takes about 1ms to complete.

## 4. Simulation and experimental results

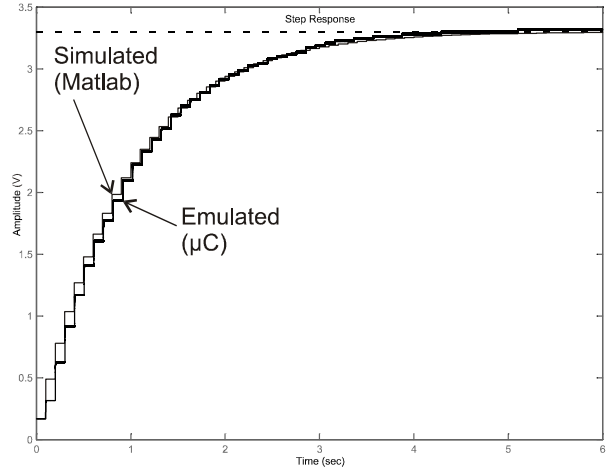
After simulation and measurements the following results were obtained:

In figure 6 is the step response of the transfer function (4) simulated in Matlab and emulated with microcontroller with 10ms sample time.



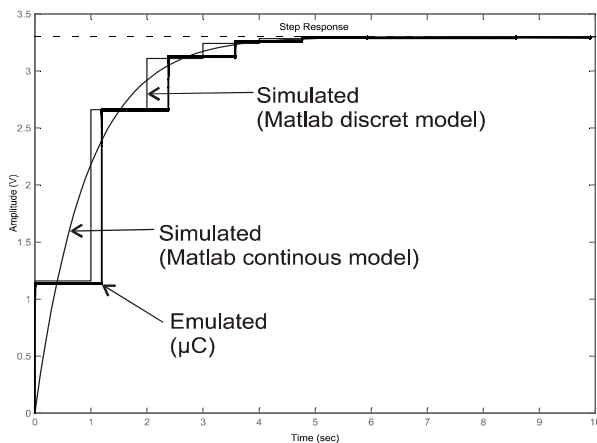
**Figure 6** Step response with 10ms sample time

In figure 7 is the step response of the transfer function (4) simulated in Matlab and emulated with microcontroller with 0.1s sample time.



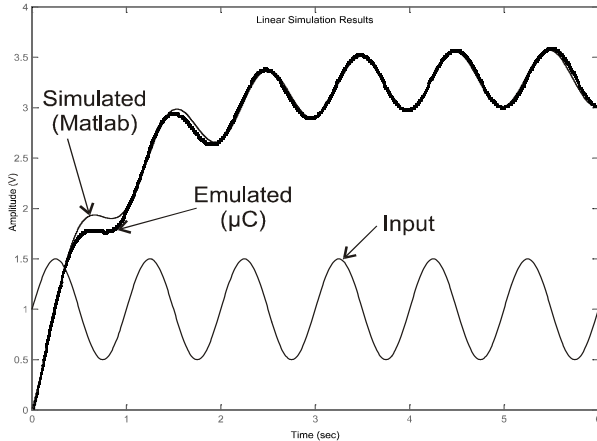
**Figure 7** Step response with 0.1s sample time

In figure 8 is the step response of the transfer function (4) simulated in Matlab and emulated with microcontroller with 1s sample time.



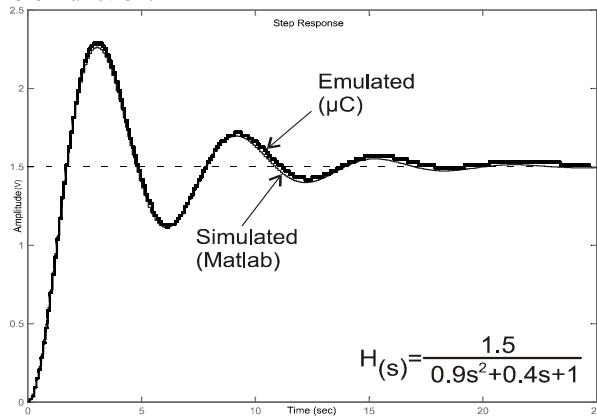
**Figure 8** Step response with 1s sample time

In figure 9 is the response at sine input of the transfer function (4) simulated in Matlab and emulated with microcontroller.



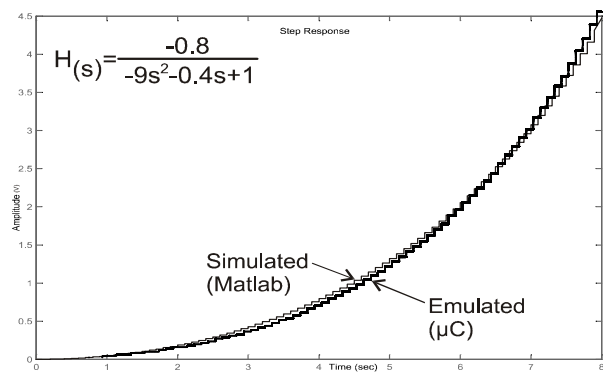
**Figure 9** Response at sine input

In figure 10 is the step response to a second order transfer function.



**Figure 10** Step response of a second order process.

In figure 11 is the step response of an unstable system.



**Figure 11** Step response of an unstable process.

## 5. Conclusions

This version of the emulator can simulate a first order and second order process with a high precision. The input and output can not have negative values. For negative values the input and output blocks has to be modified.

It can simulate unstable processes but with some limitation. The maximum value can't reach the theoretically infinite value of output because it is limited by the maximum voltage that can be generated by the output block.

For slow processes (ex: heating processes) the emulator is fairly accurate. For fast processes a faster microcontroller or a digital signal processor is needed.

## 6. References

- [1] Dogan Ibrahim, *Microcontroller Based Applied Digital Control*, John Wiley & Sons Ltd 2006
- [2] Jim Karki, *Active Low-Pass Filter Design*, Texas Instruments.
- [3] Ján Mikleš and Miroslav Fikar, *Process Modelling, Identification, and Control*, Springer-Verlag Berlin Heidelberg 2007.
- [4] Carlos A. Smith and Armando B. Corripio, *Principles and Practice of Automatic Process Control*, John Wiley & Sons, Inc. 1997.
- [5] Bonnie C. Baker, *Anti-Aliasing, Analog Filters for Data Acquisition Systems*, Microchip Technology Inc. 1999.
- [6] Ioan D. Landau and Gianluca Zito, *Digital Control Systems Design, Identification and Implementation*, Springer-Verlag London Limited 2006.
- [7] Paul M. Embree, *C Algorithm for real-time DSP*, Prentice Hall 1995