

AUTOMATIC REMOTE CORRECTING SYSTEM FOR MOOCS

Pierre-Yves Rochat, lecturer at Ecole Polytechnique Fédérale de Lausanne (EPFL)

Laboratoire de Systèmes Périphériques (LSP)
EPFL - IC -LSP, building INF, Station 14, 1015 Lausanne, Switzerland
phone: + (41) 21 693 66 45, fax: + (41) 693 66 80, email: pierre-yves.rochat@epfl.ch
web: www.epfl.ch

ABSTRACT

An automatic correcting system was designed to be able to correct the programming exercises during a Massive Open Online Course (MOOC) about Microcontrollers, followed by thousands of students. Build around the MSP430G Launchpad, it has corrected more than 30'000 submissions in 7 weeks. This document provides general information about the system, the results obtained during a MOOC on the Coursera.org platform, extensions done to remote experiences and future projects.

1. MOOC CONTEXT

Massive Open Online Courses (MOOC) are a great success from their beginning in 2012. EPFL (Ecole Polytechnique Fédérale de Lausanne [1]) is a leading university producing MOOC's, with more than 600'000 students participating in less than 2 years.

To teach embedded systems with a MOOC, there were two requirements :

- The student should have experiment boards to practice. But it is not possible for all of them.
- Grading quizzes with Multiple Choice Questions do not evaluate properly the ability of students to write working programs.

During the MOOC “Comprendre les Microcontrôleurs” (*Understand MCUs*) published on Coursera.org by Pierre-Yves Rochat and professor Jean-Daniel Nicoud [2] from October 2013, a system for remote experimentation and assignments was designed and successfully used by thousands of students from more than hundred countries around the world.

2. IMPLEMENTATION WITH MSP430

Figure 1 show the system architecture. A supervisor PC, located in a laboratory at EPFL (Lausanne, Switzerland) is connected to Internet. Two Launchpad MSP430G [3] and a serial interface are connected to the PC through USB cables.

The correcting sequence happens as follows : the student sends a file from the Coursera web pages. The file is recorded on a cloud server. The supervisor PC checks every 10 seconds if an submission is ready for correcting. A Python program is used for this, based on a script provided by Coursera.

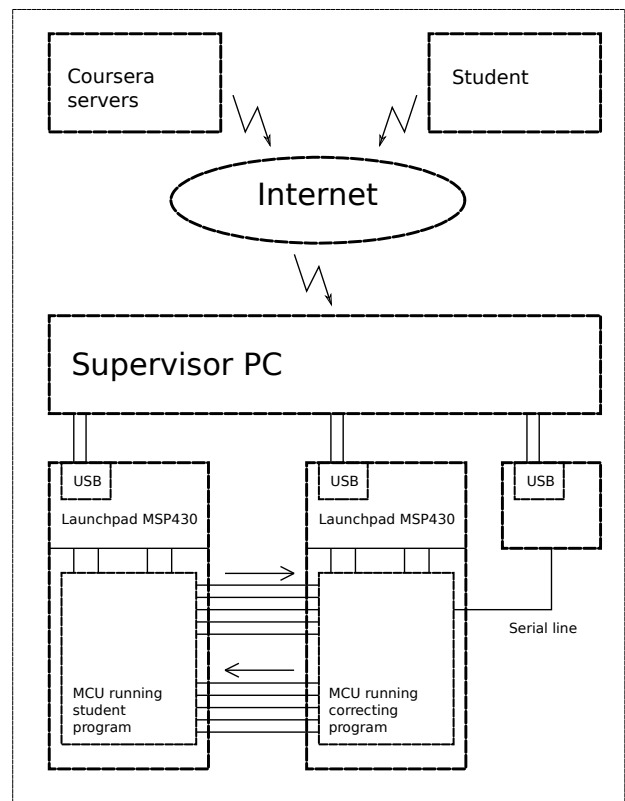


Figure 1. Architecture of the correcting system

When a program is received, the PC compiles it. GCC tool chain is used. Energia library can be use, enabling students to write normal programs, fully written, with a main() procedure, or to choose writing “Arduino like” programs, with only setup() and loop() procedures. In case of compiling errors, the error messages are send to the cloud server.

If the program is syntactically correct, the binary of the student program is uploaded on the first Launchpad. The MSP-debug utility is used. A Pearl program scans the USB ports to detect the right USB channel and transmits it to MSP-debug by a command line. The same Perl script also uploads the correction program on the second Launchpad. It is a C program for MSP430 that depends on the problem to solve.

The main correcting process can occur : the correction

program sends sequences to the input of the test program and looks at the reaction on the outputs. Protection resistors are placed between the 2 Launchpads to prevent the destruction of the output drivers of the MCU in the case of a wrong student program, which sets a pin as an output instead of an input. At the end of the test, the correction program sends the result through the serial line to the PC. A quote is joined to a small explanation, informing that the program is working properly or when a specific error has been encountered.

These results are then transmitted to the Coursera cloud server by the Python program. Finally, the student can consult the result on the Coursera web pages.

3. PROPOSED PROBLEMS

The proposed problems were simple and limited to a few inputs and outputs. To keep them attractive for the students, they were inspired from real live applications, for example :

- A simple 1 Hz blink (as first exercise).
- A Morse code SOS signal generator (the code and the standard timings were given).
- A car parking counting system with sensors for input and output cars. A “full” indication has to be generated when the maximum number of cars in the parking is reached.
- A duty cycle detector.
- A stepper motor control for both direction movements.
- An encoded lock (unlock when a specific sequence is typed on 2 keys)
- An analog level detector. The Analog to Digital Converter (ADC) of the MCU has to be used.
- A bottle filling system. A push button was used for a learning cycle, with acquisition of the filling time, another one for replication of the same filling time.
- A command of a simple mobile robot with 2 motors and 2 front sensors. The system has to avoid obstacles.

Each problem is associated with a specific correcting program, written in C and running on the second Launchpad. This program has to respect time constraints. Avoiding deadlocks is very important, since students can write surprising programs, unintentionally... or not !

With an average correcting time of less than 20 seconds, more than 180 submissions can be corrected each hour. No congestion problem occurred during the MOOC's. It is interesting to notice that many submissions came during the late evening (French speaking countries are mainly around the GMT time zone).

Compared with simulation, the system is closer to real world problems, allowing both digital and analog inputs, respecting electrical and real time constraints.

It gives a better motivation to students, as they feel “touching” real MCUs.

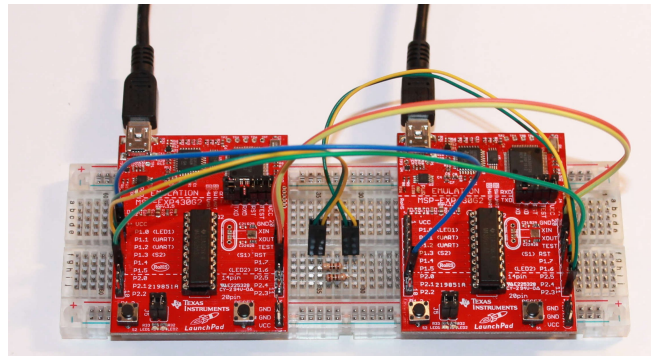


Figure 2 : Launchpads of the correcting system prototype

4. RESULTS DURING THE MOOC

During the first session of the MOOC “Comprendre les Microcontrôleurs” in October 2013, the correcting system was proposed to the students only during the last 3 weeks. Since the system was still in development, only a small part of the weekly quote relies on the single assignment of the correcting system. Figure 2 shows the 2 Launchpads of this prototype.

As we received good feedback from the students through the forum, we decided to use it more often during the second session of the MOOC in April 2014. Beside correcting programs on the Launchpad MSP430G, a version for the popular Arduino was built. The complete system is shown in Figure 3.

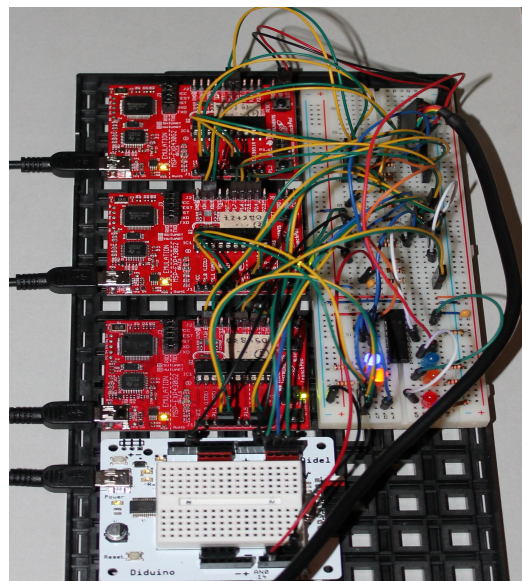


Figure 3 : correcting system of the 2014 MOOC, with Launchpad and Arduino.

Two weekly assignments were proposed to the students, forming about half of the weekly quote. For example, the command of a stepper motor and the encoded lock were the two proposed problems, giving each 10 points for a correct program, as complement of the maximum of 30 points given by the quizzes.

More than 30'000 submissions have been received during the second session of the MOOC. Feedbacks posted by the students on the MOOC's forum and emails exchanged with many of them have shown that they prefer this way of making assignments compared with traditional quizzes. The great number of submissions prove it.

5. REMOTE EXPERIMENT

Even though the students were encouraged to acquire material to become familiar with microcontrollers, it was evident that some students were using the correcting system as a distant experimenting system. Given that no limits were set to the number of submissions, they were doing many development cycles to converge to a good solution.

Form a correcting system for assignments, it became also a remote experimenting system ! EPFL (Ecole Polytechnique Fédérale de Lausanne) is very concerned in offering MOOC's to African students and universities. Distant experiment is a real added-value to online courses. This gives us a motivation to develop this aspect, for example in providing a better feedback after each submission. A temporal diagram (oscilloscope-like) would be a good approach.



Figure 4 : 16 x 16 matrix LED display. Visible in the left, a Launchpad MSP430G is used to program the flash memory

6. REMOTE MATRIX DISPLAY

During the 6th week of the MOOC, matrix displays were one of the studied subjects. To provide an opportunity for the students to program such a system, a 16 x 16 LED's display driven by a MSP430G was build and placed in front of a webcam (figures 4 and 5). After the upload of the student program, the PC records through the webcam

during 10 seconds the effect of the program on the display. Then, the recorded video is encoded in MP4 format and placed on a remote FTP server. The corresponding URL is then transmitted to the Coursera cloud server to make it available to the student as a video.

No quote was given related to this system. The students were first suggested to generate a simple pattern, the "Z" letter, the same way Zorro wrote it in the popular television films ! After this, many students wrote programs to generate nice dynamic geometrical patterns.

A library was offered to access the LED's by their X-Y addresses, but a complete description of the hardware architecture with four 64 bits serial and parallel registers could be provided to the students to enable them to write complete applications, using the "bit-banging" technique.



Figure 5 : Webcam recording the display images

7. CONCLUSION AND FUTURE PROJECTS

After the success of the system during spring 2014 session of the MOOC, we plan to improve and complete the system for the next sessions. A temporal diagram will be added to the feedback of the submissions.

New experiences with visual feedback will be proposed in addition to the matrix display such as :

- stepper motor drive
- close-loop DC motor control
- pen plotter control
- small 2 wheels mobile robots in a enclosure

Communication systems between two MCUs are not easy to experiment by students, since they often have only one target system. We plan to set a remote experience with two MCUs

connected together and programmable separately. The students will be able to experiment asynchronous serial, SPI or I2C transmission standards.

New web pages will be prepared on the EPFL web site. Contrary to the MOOC's pages, only available for students registered on Coursera and during each session of the MOOC, these new pages will be available during all the year, without any registration.

Finally, we plan to replace the supervisor PC by a Beagle Bone Black [4], a nice small computer card built around Texas Instruments chips. Such a “nano-computer” is quite powerful enough to make the corrections, with lower power needs. Given that the system works permanently, less than 200 kW are going to be used every year instead of more than 600 kWh.

GPIOs of the Beaglebone Black are planned to be used as inputs for the time diagrams acquisition, avoiding the need of a separate hardware to be connected to a traditional supervisor PC.

Many African universities are interested in providing such a remote correcting and experiment system to their students. However Internet connections are still very slow in Africa, or even sometimes non-existent. Installing local system, accessed through the Intranet of the campus (wired or WiFi) is a interesting alternative, enabling many students to work with the system without connection delay.

Since the price of the hardware for the entire system is very low (less than 200\$) , it will be easy to install the system in many countries. Contacts in Senegal, Ivory Coast, Benin, Cameroon and Congo are being established.

ACKNOWLEDGMENTS

We are grateful to EPFL for giving us the opportunity of teaching through a MOOC and the freedom to experiment a correcting system. EPFL has already a great experience in remote laboratories [5], but it was the first time when such a system was used during a MOOC

Thanks to my friends Jean-Marc Koller and Guy Maurer for the help they gave me during the project.

Acknowledgments are also given to Texas Instrument for helping us to make MCU available for many African students.

REFERENCES

- [1] Ecole Polytechnique Fédérale de Lausanne, <http://www.epfl.ch>
- [2] Professor Jean-Daniel Nicoud, http://fr.wikipedia.org/wiki/Jean-Daniel_Nicoud
- [3] Launchpad MSP430G, <http://www.ti.com/tool/msp-exp430g2>
- [4] Beaglebone Black module, <http://beagleboard.org/black>
- [5] Remote Lab of the Automatic Control Laboratory of EPFL, <http://la.epfl.ch/page-52978-en.html>