Paper ID #19320

Use of an Automated Grading Circuit for a Lab-based Course

Dr. Christopher Miller, Rose-Hulman Institute of Technology

Chris is an Assistant Professor of Electrical & Computer Engineering at Rose-Hulman Institute of Technology. His interests include engineering education, embedded systems, and ubiquitous computing.

WIP: Use of an Automated Grading Circuit for a Lab-Based Course

Abstract: Laboratories and hands-on projects are an important part of courses in embedded systems and microcontrollers. Assessment and feedback for these projects can be challenging due to the speed at which microcontrollers operate. Verification of outputs may not be possible with the human eye, and producing the necessary inputs to test all boundary conditions may not be possible through basic human interaction. Incorporating self-assessment and verification into project requirements can assist in addressing these issues, as well as helping students to develop important testing and verification skills. However, many students have yet to develop the necessary skills to understand how to properly test their designs.

We have developed an automated grading circuit for an introductory course on embedded systems and microcontrollers. The course focuses on hands-on learning through weekly lab projects and a term project. The automated grading circuit is designed to be connected to students' lab circuits, and to provide an automated test with real-time feedback and immediate results. The grading circuits are made available to students so that they may use them throughout the design and implementation process to verify their circuits and get immediate feedback prior to their final demonstration. The motivations for the development of the automated grading circuit include: improve student understanding of system verification, improve quality and quantity of feedback and enable student-directed feedback, enable more precise grading rubric criteria, and reduce demonstration time to allow more time for in-lab activities and assistance. In this paper, we present the design and implementation of the grading circuit and test programs. Additional assessment is needed to further evaluate benefits of the grading circuit and determine the best application of this tool to improve student outcomes and development.

Introduction

Laboratories and hands-on projects are an integral part of any course in embedded systems or microcontrollers. These projects allow students to explore the connection between software and hardware, and to become familiar with the complexities of all aspects of implementation, from design to verification. Assessment and feedback for these projects can be challenging due to the speed at which microcontrollers operate. Verification of outputs may not be possible with the human eye and may require lab equipment. This is time consuming, and may not be feasible in time-limited lab sessions. Additionally, even if observation of outputs is possible, it may not be possible to manually produce (through human interaction) the necessary inputs to test all boundary conditions. These issues present challenges to timely and detailed feedback, which is important for student learning[1,2,3,4,5,6].

Incorporating self-assessment and verification into project requirements can assist in addressing these issues, as well as helping students to develop important testing and verification skills. While testing and verification may not be explicitly listed among course objectives, it is an important skillset that must be learned throughout the curriculum[7,8] for all electrical and computer engineering majors. However, in introductory courses, many students have yet to develop the necessary skills to understand how to properly test their designs. They, therefore, remain unaware of shortcomings in their understanding until feedback is received, at which point they may have moved on to future laboratory assignments. In addition, assessment and feedback must still be provided for both the design and implementation as well as the testing and verification. If the testing and verification is inadequate, additional methods are needed for assessing the design and implementation.

To address these issues, we have developed an automated testing circuit for laboratory circuits in an introductory course on embedded systems and microcontrollers. The automated tester can provide the detailed stimulus necessary to assess the accuracy of the implementation. In addition, by making the tool available to students, they may use it to assist in their testing and verification, and it can serve as an educational tool on proper verification methods. The motivations for the development of the automated testing circuit include: improve student understanding of system verification, improve quality and quantity of feedback and enable student-directed feedback, enable more precise grading rubric criteria, and reduce demonstration time to allow more time for in-lab activities and assistance. In this paper, we present the design and implementation of the grading circuit and test programs.

Methodology

We have developed an automated grading circuit for a sophomore-level, introductory course on embedded systems and microcontrollers. The automated grading circuit is designed to be connected to students' lab circuits, and to provide an automated test with real-time feedback and immediate results. The grading circuits are made available to students so that they may use them throughout the design and implementation process to verify their circuits and get immediate feedback prior to their final demonstration.

Course design

The introduction to embedded systems and microcontroller course focuses on hands-on learning through weekly lab projects and a term project. It is a 10-week course as part of the quarter system. Labs do not utilize pre-constructed test boards. Instead, students construct their own breadboard-based circuits for each lab. Students purchase a lab kit at the beginning of the term, which includes all of the necessary components for the labs including the 8-bit Microchip PIC16 microcontroller. An overview of the topics covered throughout the course are provided in Table 1. The course begins with an introduction to the instruction set architecture (ISA) and assembly programming. The first three labs are therefore completed using assembly language. The remaining labs are implemented with the C programming language, which is introduced in week three. An overview of the course labs is shown in Table 2.

Table 1. Outline of course.

Week	Topics
1	Introduction, software development tools, and PIC microcontroller architecture
2	PIC instruction set, hardware timers, internal pull-up resistors, basic I/O and mechanical switch debounce methods
3	Programming in C, timers, and compare modules
4	Hardware timers with interrupts, compare/capture modules, and driving high-current loads
5	Servo motors, pulse width modulation, analog-to-digital conversion, and interfacing with LCD displays
6	RS232 Universal Asynchronous Receiver-Transmitter (UART) serial communication, framing and parity, and interfacing with keypads
7	Inter-integrated circuit (I^2C) synchronous serial communication, temperature sensors, and stepper motors
8	Embedded system design principles and advanced topics
9	Final project design and modern development tools
10	Final project completion

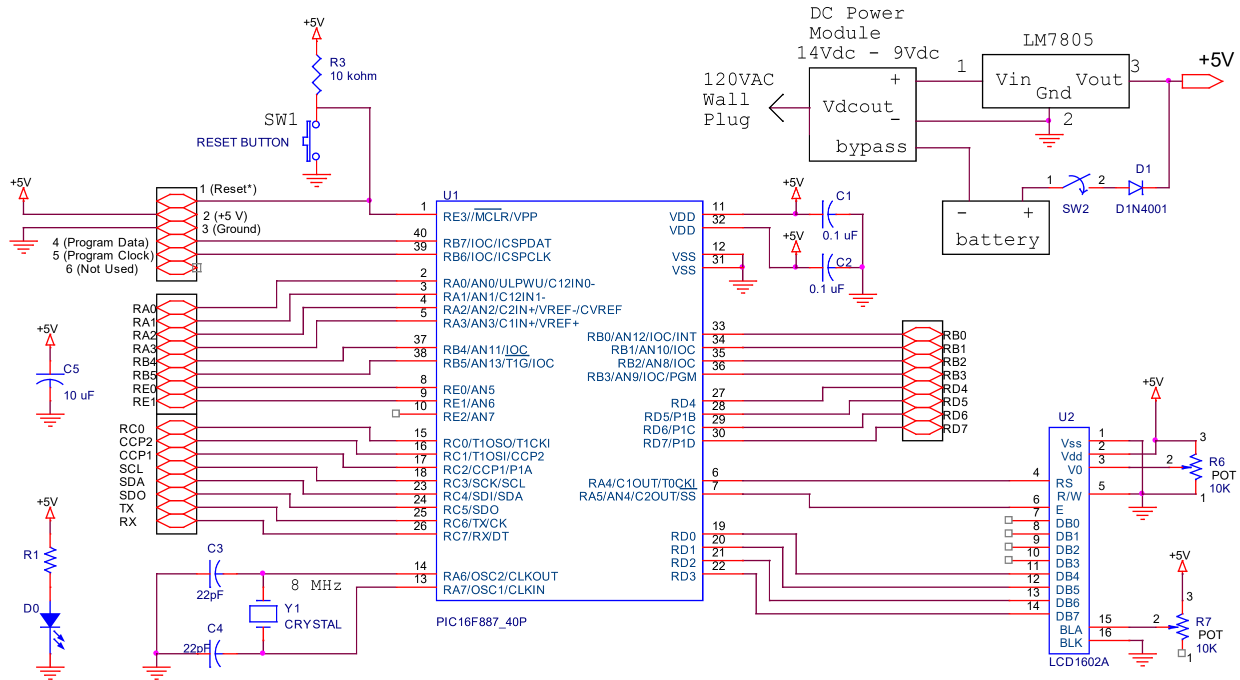
Table 2. Example of course labs.

Lab	Title	Description
1	Pushbutton and LED	Familiarize students with lab kits, microcontrollers, development software, assembly programming and basic circuit concepts. Timing delays – fixed-length instruction loops.
2	Double-click detector	Familiarize students with testing/debugging tools. Introduce variable data storage, timers and de-bouncing concepts. Timing delays – timer with overflow flag.
3	LED quick reaction game	Introduce 16-bit timers and compare modules. Introduce function calls and parameter passing in assembly. Timing delays – timer with compare module.
4	Music note player and frequency tuner	Programming in C. Introduce capture module and compare with output effect. Introduce interrupts. Introduce potentiometer, speaker, nMOSFET, square-wave signals. Timing delays – timer and compare module with interrupts.
5	Light sensor with servo	Introduce analog-to-digital conversion. Analog input sources. Introduce LCD display, PWM and servos.
6	Remote combination lock system	Introduce UART and serial communication – baud clock generation, data framing, and parity. Introduce keypad input.
7	Temperature control system	Introduce I^2C serial communication – data framing, addressing. Introduce stepper motors.

Design of the automated tester

The automated tester was designed using the same components available to the students in their lab kit. This has the effect of letting the students know that the design and implementation of such a tester is within their capabilities. Additionally, I provide the schematic of the tester to the students so, if two students who are partnered want to construct their own test circuit with one of their lab kits, they are able to do so. The test circuit was designed to provide a generic interface that could be suited to many different labs so that only a single test circuit design is used. A different test program may be loaded on the circuit for the appropriate lab. Due to program memory limitations on the microcontrollers in the student lab kits, only a single test program can be stored. However, if a microcontroller with larger program memory capacity were used, all of the test programs could be included and user-selectable. The circuit schematic is shown in Figure 1. The tester circuit incorporates an 8-pin header for a 16-button keypad and an LCD as shown on the right side of the diagram. The circuit exposes 16 I/O pins, shown on the left side of the diagram, which may be used for generating test stimulus or measuring lab outputs. Additionally, a 6-pin header is provided for connecting the PICKit programmer, as shown in the upper-left of the schematic.

Figure 1. Circuit schematic for automated tester circuit.



Test programs

Each lab has a separate test program specific to that implementation. A test procedure is determined for each lab, which serves as the basis for the program design. The set of input/output connections necessary are individually determined for each lab based on the number of stimulus signals needed as inputs for the lab circuits, and the number and type of output signals which must be captured and/or measured. The programs produce the necessary outputs which will serve as stimuli for the lab circuit (such as emulating a button press), and capture the value or timing of output signals from the lab circuits. For each lab, students are provided with a test circuit document, which details the connections that must be made, describes the output parameters, and the test procedure that is used. Inclusion of the test procedure is intended as a learning tool for students, to help them understand how a test can be designed to properly verify that a circuit and program meets specifications. An example of the test procedure for a simple LED toggle project is shown in Figure 2. An example of the complete test circuit document for Lab 3 is provided in the appendix. The automated tester circuit is particularly useful for testing the quick-reaction game implementation, as it allows us to generate inputs at precise timings to verify all boundary conditions in the lab implementation.

Figure 2. Test procedure for Lab 2 – Double-click detector.

<u>Test procedure:</u>

1. Initiate press of input button (active-low signal)
   i. Toggle signal on input button for 4 ms to emulate switch-contact bounce
      a. Verify LED1 remains off to indicate debouncing implemented
2. Delay 50 ms then emulate release of button (set input signal high)
   i. Verify that LED1 turns on immediately following release of button
   ii. Measure time that LED1 is on – verify that it is exactly 1 second
3. Wait 500 ms then check LED1 and LED2 state to verify score
4. Repeat process for alternate combinations of a second button press and release
   i. No second button press
      a. Test accurate score for single press
   ii. Second press at 400 ms, release at 600 ms
      a. Test accurate detection of press and release within 1 second
   iii. Second press at 900 ms, release after 1 second
      a. Test accurate detection of press and release overlaps 1 second
   iv. Second press at 975 ms, release after 1 second
      a. Test accurate detection of press and release overlaps 1 second
   v. No second button press
      a. Test accurate resetting of button press flag for each trial

Test circuit programs have been developed for labs 2, 3, 4, and 7. Lab 1 is a simple LED blinking circuit to familiarize students with the lab components and development tools and does not require verification. Lab 5 is complicated by the need for producing analog stimulus signals. This could be done, but would require equipment outside of what is currently provided in the lab kits. Therefore, other methods of assessment and verification were used. For Lab 6, the UART communication is complicated by the varied implementations resulting in unpredictable character sequences. However, a more precise project specification would allow a test program to be designed for this lab, and a testing circuit would be a useful addition to this lab.

Conclusions and future work

Introduction of the automated testing circuit in the course has, thus far, been a useful experience. It has allowed much more precise assessment and verification of embedded systems, particularly for those with significant timing requirements. Having the testing circuit available to students has led to improved preparation for demonstrations. Students are generally able to identify before demonstration if their lab circuit is working properly or if there are issues that need to be addressed in order to meet specifications. For some labs, such as the servo mounted light-detector circuit, we found that it was easier to verify functionality through visual observation than with a testing circuit, but we continue to look for additional ways in which the testing circuit may be incorporated into all lab verifications.

It is our hope that the testing circuit serves not only as a valuable tool for instructors in assessment, but for students in their learning and development, particularly their development of testing skills. We have made the testing circuit available to students throughout the term and have included the test procedure in the documentation so that students can learn how test procedures are designed to match project specifications. Further work is needed to continue to integrate this learning process in the course and to develop methods of assessment of student learning of test and verification methods.

References

1. Richard Higgins, Peter Hartley, and Alan Skelton. "The conscientious consumer: Reconsidering the role of assessment feedback in student learning." *Studies in higher education* 27.1 (2002): 53-64.
2. H. Sondergaard and Doreen Thomas. "Effective feedback to small and large classes." In *Proceedings of the 34th Annual IEEE Frontiers in Education (FIE)*. 2004.
3. Dylan Wiliam. "Assessment: the Bridge between teaching and learning." *Voices from the Middle* 21.2 (2013): 15.
4. John C. Chen, Jennifer Kadlowec, and Dexter Whittinghill. "Work in progress: Combining concept inventories with rapid feedback to enhance learning." In *Proceedings of the 34th Annual IEEE Frontiers in Education (FIE)*. 2004.
5. Jamie Cromack. "Technology and learning-centered education: Research-based support for how the tablet PC embodies the Seven Principles of Good Practice in Undergraduate Education." In *Proceedings of the 38th Annual IEEE Frontiers in Education (FIE). 2008*.
6. John C. Chen, Dexter C. Whittinghill, and Jennifer A. Kadlowec. "Classes that click: Fast, rich feedback to enhance student learning and satisfaction." *Journal of Engineering Education* 99.2 (2010): 159-168.
7. Terry Shepard, Margaret Lamb, and Diane Kelly. “More testing should be taught.” *Communications of the ACM* 44, 6 (June 2001), 103-108.
8. Stephen H. Edwards and Manuel A. Perez-Quinones. “Web-CAT: automatically grading programming assignments.” In *Proceedings of the 13th annual conference on Innovation and technology in computer science education* (ITiCSE '08). ACM, New York, NY, USA, 328-328. 2008.

Appendix

Lab 3 testing circuit

Test procedure:

1. Initiate press of start button (active-low signal)
    i. Toggle signal on input button for 4 ms to emulate switch-contact bounce
        a. Verify LED1 remains off to indicate debouncing implemented
2. Delay 50 ms then emulate release of button (set input signal high)
    i. Verify that LED1 turns on immediately following release of button
    ii. Measure time that LED1 is on – verify that it is exactly 250 millisecond
    iii. Measure time that LED1 is off – verify that it is exactly 250 millisecond
    iv. Measure time that LED2 is on – verify that it is exactly 250 millisecond
    v. Repeat for remaining LEDs
3. Wait 250 ms then check LEDs to verify score
4. Repeat process for alternate combinations of play button press and release
    i. No play button press – test accurate score for miss (no press)
    ii. Play button press under 10 ms, release at 45 ms
    iii. Play press at 45 ms, release at 90 ms
    iv. Play press at 54 ms, release at 145 ms
    v. Play press at 97 ms, release at 160 ms
    vi. Play press at 102 ms, release at 205 ms
    vii. Play press at 149 ms, release after 250 ms
    viii. Play press at 155 ms, release at 195 ms
    ix. Play press at 195 ms, release after 250 ms
    x. Play press at 201 ms, release at 245 ms
    xi. Play press at 249 ms, release after 250 ms
    xii. Play press after 250 ms, release after 250 ms
    xiii. Play button press and release too early (before 4^{th} LED)
    xiv. Play button press too early release during 4^{th} LED blink
    xv. Play button press too early release after 4^{th} LED blink

Scoring:

DB: switch debounce score [0-6]
TM: LED blink timing score [0-24]
CD: scoring accuracy [0-30]

// ERROR CODES
// 1 - LEDs do not turn off after press of button
// 2 - LEDs do not turn on to indicate completion/result of trial
// 3 - LEDs do not remain on until button press of next trial
// 4 - score output before minimum delay
// 5 - LED does not turn on in sequence
// 6 - LED does not turn off after 250ms delay

Grading circuit connections:

Pin	Signal	Grader circuit	Lab circuit
1	Start	RC0	RB0
2	LED1	CCP2	RD0
3	LED2	CCP1	RD1
4	LED3	SCL	RD2
5	LED4	SDA	RD3
6	Player input	SDO	RB1

* remember to also connect Grounds