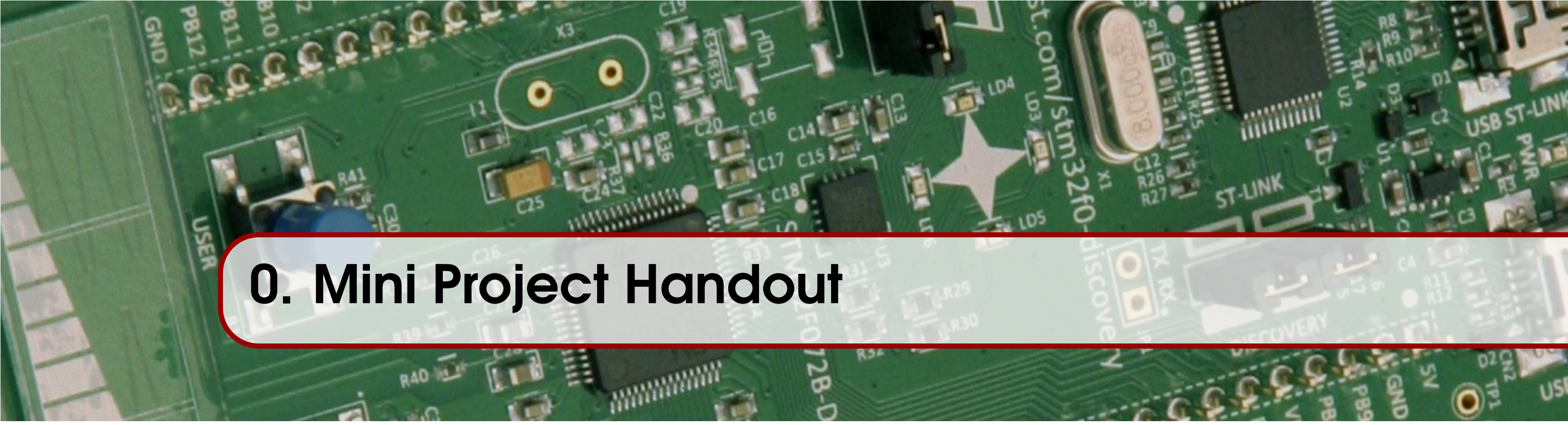

0. Mini Project Handout

0.1 Introduction

The mini-project is a chance for you to not only demonstrate what you've learned in the class, but to use your imagination and personal interests to investigate something that appeals to you. Just keep in mind that the mini-project will only cover a couple of months, so it will have to be scaled or structured accordingly. This does not mean that your dream project must be completely finished by the deadline, only that the part of your project we decide is reasonable in your proposal is completed. For example, many students are already expressing interest in using brushless DC motors to create drones, cars, or stabilization devices. Just the acquisition and assembly of these things can possibly take you weeks! So instead we might recommend that you design a motor driver library in C as your project, or interface an existing driver with the Discovery's gyroscope, or an accelerometer. Once the requirements outlined in your proposal have been met, you can focus on the rest of your project.

0.2 Mini-Project Timeline

Your mini-project will consist of the following milestones:

1. Project Proposal
2. Milestone 1 Checkoff
3. Milestone 2 Checkoff
4. Milestone 3 Checkoff
5. Demo
6. Final Delivery

The milestone checkoffs will be scheduled during your regular lab times, and the demo will happen during regular lab times in the last lab sessions of the semester.

0.3 Proposal Submission

You must provide a written project proposal. The proposal should contain:

1. List of the group members (Up to 4 group members – the larger the group, the more strenuous the milestones should be)
2. High-level description of your application – 4-5 lines of text
3. Functional block diagram, containing all the hardware/software blocks of your project.
4. A list of 4 implementations milestones. The 4th milestone is due with the final delivery (after finals week).

Your proposal will be reviewed for practicality and merit by the TAs to ensure the success of your mini-project. You must meet with your lab TA to discuss the project idea before the deadline and have him check-off the general technical solution.

The mini-project can be picked from the list below, or may be a custom project. These projects are calibrated to fit into the short period of time that you have. However, we value the innovation and will be happy to approve (and reward through the “innovation” criteria of the grading rubric shown below) a custom project of your own. It is OK to dream and think big to start with. Consider several ideas if you like. Be sure and discuss your ideas with your group and the staff early.

0.4 Project Ideas

Minimal Complexity

- Interface on-board gyroscope with motor
- Motor speed control with speed value set by PWM input duty cycle
- Motor position control instead of speed
- Stepper motor driver
- Develop non-blocking driver for USART, link with printf

More Difficult (and more rewarded!)

- MPPT solar tracker
- Star tracker
- 3-Axis Gimbal
- Addressable LED driver
- Automated coffee machine
- Control a relay and stay accurate with RTC (32KHz) baseline
- 2-wheel balancing robot
- Persistance-of-Vision LED Propeller
- Russian roulette using multiple boards and multi-master I2C
- MIDI Controller / Soundboard
- Line follower vehicle
- Obstacle avoidance vehicle
- Autonomous vehicle
- Audio Gaphic Spectrum Analyzer (May want to use an F3 or F4)
- Oscilloscope or Logic Analyzer
- WiFi-enabled smart outlet
- Infrared communication between boards
    - Half-duplex
    - Full-duplex
    - Interference mitigation

0.5 Custom Projects

Instead of picking up a project idea listed above, we encourage you to propose your own project (and get more innovation points). You can use the above ideas to create your own project, and innoate it as you’d like! Here are a few tips that might spark some creativity as well:

- You are not limited to the Discovery as your sole MCU! Interface with an Arduino, Raspberry Pi, or other microcontroller. We only require you to utilize the MCU's low level libraries to implement at least 3 of the labs (interrupts, timers, UART, etc.) For example, your entire project cannot be written in C++, Python, or high-level C APIs, but as long as **at least 3 labs are implemented using low-level C functionality** then go crazy with the rest!
- Connect a simple Bluetooth interface to your project with a USB Bluetooth dongle and a USB Host Shield. You can easily connect a PlayStation controller or create a simple application on your smartphone to send commands to your Discovery board.
- Use a ESP8266 to cheaply add WiFi to your project. Purchase something like an Adafruit Feather HUZZAH with ESP8266, and easily create an I2C or UART interface to your STM32 by programming the HUZZAH in the Arduino environment. You can then connect to a local WiFi network (like your laptop), create a socket, and wirelessly communicate with your Discovery.
- An IMU (*Intertial Measurement Unit*) can add a lot to a project. The combination of gyroscope and acceleromter can take your balancing robot or autonomous vehicle to the next level.
- Look at websites like Adafruit, Sparkfun, and Hackaday for project inspiration.

Please consider the following restrictions for your project:

Safety Restrictions:
Several safety issues must be in mind when dreaming about your project.

- **Spinning Devices:** High speed spinning mechanics must be contained so that in the event that they fail that they do not throw debris. This can cause eye injury and we must be very careful. You may also be required to wear safety glasses when operating this equipment.
- **Consumable Foods:** Food oriented projects that involve beverages or food have to avoid contact with toxic substances like lead based solder or toxic adhesives.
- **Alcoholic Beverages:** University Policy Prohibits use of alcoholic beverages on University Property including non-alcoholic beers.
- **Projectiles:** Projects involving projectiles must use soft materials and restrict velocities. Some form of containment might be required.
- **Heat:** Projects involving heat may require some form of insulation or isolation. High heat levels that can cause combustion will not be allowed.
- **High Voltage:** Projects requiring voltages like the line current from AC outlets must us special isolation devices available in lab.
- **Lasers:** Project using lasers must provide containment or shielding to prevent light from reflecting and potentially entering any ones eyes.
  Note: You may not proceed with project components that involve safety issues until approved by the staff.

Project Size:
Be sure and consult the staff about projects requiring a great deal of lab space. Generally, we can accommodate you if it does not interfere with other projects and workstations. Since the kits lend themselves to mobile applications, it is possible to use areas outside the lab, but you should check with staff about using public areas such as the halls or atrium.

Requirements:

In order for your project to get full points, make sure to know the expected requirements and study the grading rubric.

0.6 Grading Rubric

The project will be evaluated using the following set of criteria.

Table 1: Summary

Category	**Maximum Value**	**Good Project**
Base Requirements	15%	15%
Successful Milestones	20%	20%
Difficulty Component	15%	10%
System Integration	15%	12%
Innovation	6%	2%
Extraordinary Technical Achievements	4%	1%
Proposal	20%	18%
Documentation	5%	5%
Max Total	100%	83%

Base Requirements:
Your project or application should demonstrate your basic laboratory skills with at least 3 of the primary lab topics:

1. Modeling
2. Bus Interfacing (serial device such as UART or I2C)
3. Interrupts
4. General Purpose Timers: Timers/Counters/Inputs as timers
5. Analog to Digital (ADC) or Digital to Analog (DAC) Conversion

Successful Milestones:
In your proposal, you have to list 3 intermediate milestones to acheive during your project. Every successful milestone will grant you 5%.

Difficulty Component:
Solving an application problem apart from what you have experienced in the lab, will challenge your laboratory skills and knowledge of the course concepts. For example, by the time you complete the last lab, it should not be very difficult to use basic IO on the STMDISCOVERY kit and convert analog signals to digital values you can use in a software application.

On the other hand, if you need to use external devices such as displays, sensors or actuators, you would be facing a host of unknowns. While at first this task may seem formidable, you actually have all the skills and knowledge to tackle this challenge. You begin by obtaining and reading the product specification. In the serial interfacing lab, you learned how to communicate with external devices. The specifications provide you with a functional overview; power supply requirements serial timing requirements and functionality of the interface registers. So, while you are dealing with new challenges, you are in fact applying concepts and skills you have tried at least once in the labs. Applying your knowledge and skills to an application category apart from what you have seen in the lab, defines the difficulty category. Difficulty is scored from 0 – 35%. Consider the following examples:

- **Easy (0%):** Using STMDISCOVERY Simple IO like switches and LEDs. Generally, anything that is a simple extension of the lab exercise does not earn difficulty value.
- **Moderately Difficult (1% - 10%):** OLED display on STMDISCOVERY kit. While this is extension of one of the labs, there is some difficulty in adapting to a new application.
- **Difficult (11% - 15%):** Difficult device application requiring significant work. Perhaps, custom printed circuit, additional signal conditioning such as filters, amplification or current buffering. This type of application usually requires a great deal of device characterization. For example, a I2C interfaced power monitoring device requiring a custom printed circuit board and characterization of inductive pickup. We are also lenient when it comes to mechanical components. This is not a mechanical engineering lab, so we do not expect mechanical devices to work perfectly. We expect you to do your best matching a mechanical component to your application, but if a component falls off during a demo because the double-sided sticky tape failed we will not consider this a great failure.
- **Difficulty Component Weighting for Group Size:** Learning to work in a group is an important part of your project experience. In industry, it is not uncommon to have several groups of 3-5 persons supporting a project. As a group, you will have to decide how to distribute your tasks and cooperate to integrate each part. Typically, a project group consists of your semester's lab partners; however, it is permissible to form alternate groups. In fact, you can form larger groups if you wish. Larger groups can produce some great projects because of the additional resources. In fact groups of three can be very effective and easily offset the weighting factor. You may work alone if you wish; however, you will need the permission of the course or lab instructor. To compensate for different groups sizes, the difficulty portion of the projects will be weighted accordingly (but capped in the 15% limit):

Group Size	Weighting Factor
1-2	1.4
3	1.2
4	1.0

System Integration:
At some point, you will have to integrate the project components into a functional application. It is quite possible that your system components work well independently, but do not work to serve your application. You should be able to answer the question: Do the components of the project work as a cohesive whole? Specifically, consider the following:

- Are the components relevant to the application?

  - Do the sensors support the application or are they superficial?
  - Is display information relevant?
- Is component performance relevant and appropriate to the application?
  - Sensor Range/Response?
  - Actuator Range/Resolution?
  - Display Capacity/Response?
- Is the application reliable and repeatable?
  - Is it necessary to reset the entire system after each trial?
  - Do mechanical parts fall off?
  - Do connections break?
  - Do components require physical adjustment after each trial?
  - Do electrical components (potentiometers, etc) require adjustments between trials?
- Is system function appropriate for application?
  - Is the system responsive?
  - Functional ranges adequate?

For example, consider our simple motor speed controller. A sensor and motor control circuit can be implemented to sense and control the speed of the motor with a traditional feedback path. It is possible that the display, motor speed control and speed sensing all work independently, but the system does not function well as a whole. For example, does the motor controller system:

- Hold motor speed as a function of load?
- Respond to load changes fast enough?
- Display the motor speed accurately?

This factor measures the success of your application. Keep in mind that refinements can be very time consuming. It is better to address the significant functional components first. For example, a motor controller that does a good job of providing regulation over a limited load range is better than one that works crudely over a broad load range. Or, nuances in the display function are insignificant if the application performs poorly. General Evaluation Criteria for System Integration: 1. Components do not work well together. (0 points) 2. Some components work together while others do not. (4 points) 3. Most components work together. (7 points) 4. All the components fundamentally work together, but under some conditions the system experiences minor failures. (10 points) 5. The system works well enough you would be satisfied buying it. It is responsive, fast, does something interesting, bug free, etc. (15 points)

Innovation:
We will assign an innovation factor to projects, which exhibit unique, interesting attributes, beyond the subjects proposed later in this document. While this factor is subjective, the best way to define it is to consider the response of an unbiased observer. For example, if freshmen engineering student's first words after seeing your project were "that's cool", the project is probably cool. We have all experienced this. Just because an application solves a significant or representative engineering problem does not make it "cool". Past projects with coolness attributes are Spider, Rubik's Cube Solver, Servo Climber and several others. Look them over and see what you think. Keep in mind that you will have less time than previous years, so the scope of your project may not be as broad.

General Evaluation Criteria for Innovation:

1. A high school senior might make a casual remark like “hey, it’s pretty cool”. (0 points)
2. A high school senior would find it interesting. (2 points)
3. A high school senior would find it remarkable. (4 points)
4. A high school senior would tell talk about telling their friends for weeks. (6 points)

We will use this criterion to reward custom projects.

Extraordinary Technical Achievements:
Occasionally, a group will accept and overcome unexpected significant technical challenges. Students in this category are often interested in a particular area and surprisingly overcome application challenges beyond the scope of the course. Merit in this category is rare and may only amount to a 1 or 2.

General Evaluation Criteria for Extraordinary Technical Achievements:

1. Project meets requirements, but no major technical hurdles were overcome. This is typical of most projects. (0 points)
2. The project includes significant technical achievements of one sort or another. For example, a non-trial device never before used was used in a challenging way. (1 point)
3. The staff thinks there were huge technical hurdles to overcome. This is rare! (3 points)
4. Technically nearly impossible. These are rare also and may not occur for several semesters. (4 points)

Documentation:
Adequate documentation for both hardware and software is expected. Appropriate function descriptions, lists and descriptions of parts used, etc.

0.7 Milestones Checkoff

Your milestones will be checked off by a TA during your lab sessions.

0.8 Hardware and Software Archive

This goes along with Documentation. You will have to provide a final version of your hardware and software (Schematics, PCB design folders, C project, etc.).

0.9 Balancing Your Efforts

You can score well by balancing your efforts over the project categories. The project scoring is designed to encourage you to produce a project with sufficient number of difficulty components to successfully address your application and provide good system integration. Consider the following to balance your efforts:

- **Difficulty Components:** Choose devices and functionality that supports your application.
  - Displays: Use a display that supports your application: do you need graphics or just relevant text?

  - Sensors: Use sensors that support your application. Do they have the range, resolution, dynamic response?
  - Actuators: Use actuators that support your application. Do they have the range of motion, precision, power, etc?
  - Manual Input: Do you need a joystick or simple button matrix?
- **Number of Difficulty Components:** Do you have enough or too many for your application?
  - Your application may have required another sensor, actuator etc to improve better control and system integration.
  - You may have sensors or actuators that do not help your application when your time would be better spent on system integration.
- **System Integration:** Make sure you allocate time to insure yours device and functional components work well together.
- **Group Size vs Difficulty:** Make sure your application is on par with your group size.
  - Are there enough members to support the application?
  - Is the application sufficiently difficult for the group size?
  - Most advanced project require 3 to 4 people
- **Do the Essentials:** Be sure and spend some time on the easy points:
  - Hardware/Software Points: Get it in on time
  - Proposal: Get it in on time
  - Base Requirement: Usually easy