Lab 4: Heading/Ranger Integration, Battery Voltage and LCD Display

Preparation

Reading

Lab Manual

Chapter 4 - The Silicon Labs C8051F020 and the EVB (Analog to Digital Conversion)

Chapter 5 - LCD and Keypad

Chapter 7 - Control Algorithms

LMS, Course Materials

Electric Compass Data Sheet

Ultrasound Ranger Data Sheet

LCD and Keypad Data Sheet

Wireless RF Serial Communication Link

Objectives

General

1. Integration of the compass and ranger systems developed in the previous lab to control the steering and driving. The terminal emulator (SecureCRT or PuTTY) through the wireless serial link will be used to collect data, set some options in the program and set the speed.
2. Use Analog to Digital Conversion as was implemented in Lab 2 to read an A/D input
3. Display control information on LCD display screen or the terminal and enter variables using either the keypad or the terminal keyboard.

General Specifications

1. Start on the west side of the paper (near the step).
2. Have the slide switch in the 'off' position.
3. Enter the program parameters using the keypad.
    a. Set the desired direction to east +/-30^{o} (600 to 1200 tenths of a degree)
    b. Set the proportional gain for the steering and the drive motor.
    c. Check the potentiometer for the voltage input.
4. Move the slide switch to the on position
5. Set the BiLED to green.
6. Start driving forward
7. Start recording data (saving to a file or displaying on the putty terminal).
8. Drive forward for at least 3 seconds.
9. Repeat
    a. Steer in the desired direction, ~east if going forward, ~west if going in reverse.
    b. Start checking the ranger for an object within 20cm.
    c. If an object is detected,
        i. Reverse the direction of the drive motor (forward to reverse or reverse to forward)
        ii. Set the BiLED appropriately.
        iii. Wait 3 seconds before starting to check the ranger again.
    d. If the slide switch is set to the off position, go to step 3.

More details in the following sections.

Hardware

1. Wire a single protoboard with an Ultrasonic Ranger, Electronic Compass, and Keypad/Display. Only one set of pull-up resistors is needed on the SMB (I2C Bus).
2. Design a voltage divider circuit to control the car speed and display the speed as a percentage of maximum speed.
3. Add a wireless RF serial transceiver to replace the serial RS-232 wired connection.
4. Add a slide switch that can disable (set to neutral) the drive motor and the steering servomotor so these can be turned off while making adjustments to the car while the program is still running.
5. Add a BiLED to indicate the direction the car is driving (green for forward, red for reverse).

Software suggestions

1. Integrate the C code used in Lab 3.
    a. Write a function that calls a read_compass() function that reads the compass direction and sets the PWM for the steering servo based on the present heading and a desired compass heading. Note, both the compass reading and the steering change/correction should be only be completed when new data is available on the compass
    b. Write a function that calls a read_ranger() function and determines the proximity of an object. The range **will not** be controlled by the ranger reading. Note, the ranger should only be read when new range data is available.
2. Integrate code from Lab 2
    a. Write C code to configure the A/D converter in the C8051 to read the voltage input from the potentiometer. Use your code from Laboratory 2 as a template and choose an input on Port 1.
3. Control algorithms
    a. The steering control should be a linear map using the error in direction (difference between current direction and desired direction), as indicated in item 1 above.
    b. The speed control should be a linear map using the A/D conversion value of the potentiometer voltage. The drive motor proportional gain minimum should correspond to an input voltage of 0V producing a neutral pulsewidth and an input voltage of 2.4V resulting in maximum speed forward or reverse. The drive motor proportional gain term can be increased, such that maximum speed occurs at an input voltage less than 2.4V. Neutral should always correspond to 0V.
4. Inputting parameters
    a. Write a UI (User Interface) program that allows the user to input
        i. A proportional control term for the steering
        ii. A proportional control term for the speed
        iii. The desired direction
5. Saving data
    a. Write formatted print statements to display
        i. The current direction (compass reading)
        ii. The direction error
        iii. The steering pulsewidth
        iv. The current distance (ranger reading)
        v. The ADC1 value (only needs to be read at the start and when the car reverses direction.)
        vi. The drive motor pulsewidth

Code merging

For this lab and for the rest of the course, the team of 3 students is expected to do some parallel development of the required functions needed to complete the labs using the assigned protoboards with simple test circuits. The individual members should have specific tasks to accomplish for the lab.

The hardware and software from Lab 3 should be merged if not done so already. The team must be careful when merging the software to make sure that initialization functions and variable usage is consistent with the task. The initialization of the PCA is one area where changes may be required. Both sensors use the SMBus, which is valid. Many slaves can be on one SMBus, and both sensors, as well as the LCD/keypad, all function as slaves.

The integrated software and hardware will result in a car that starts in the indicated position on the west side of the paper, near the step. Without modifying the Lab 3-3 codes, the car will drive at a a speed dependent on the ranger value and will follow a compass heading of east.

While running, the integrated software must also continuously poll the run/stop slide switch connected to P3.7 to turn on/off (set to neutral) both the drive and servomotors. Other details are left up to the team.

Hardware

The LCD/keypad should mount on the back of the car as shown below. The board has a 4-wire cable that connects power, ground, SDA and SCL. The LCD board must be left on the car or returned each class and not kept with your protoboard. The RF transceiver module should be returned to the podium table.

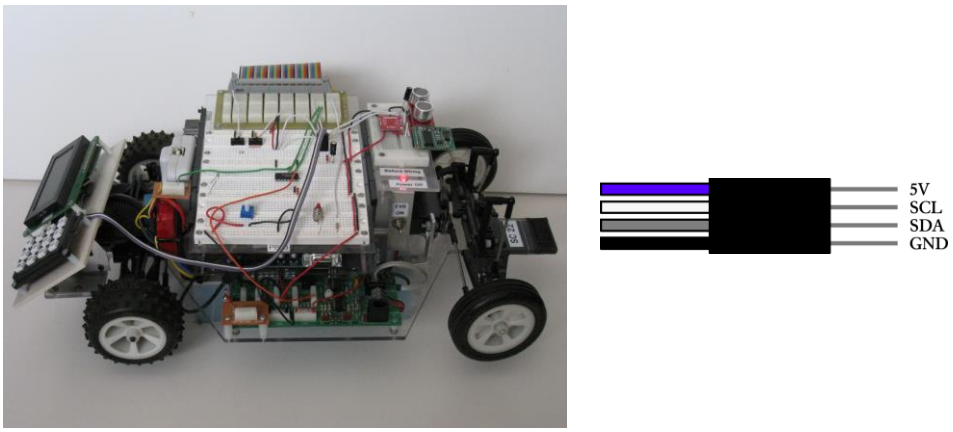

Figure 4.1 - Mounting of LCD screen on Smart Car (left) and LCD connector pinout (right)

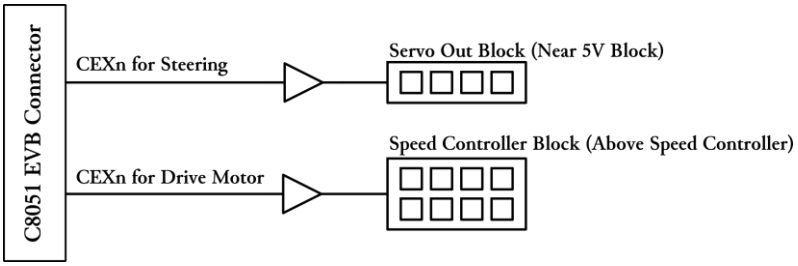

Figure 4.2 - Car steering and drive motor control circuitry from Lab 3 (part 1)

System Management Bus (SMB)

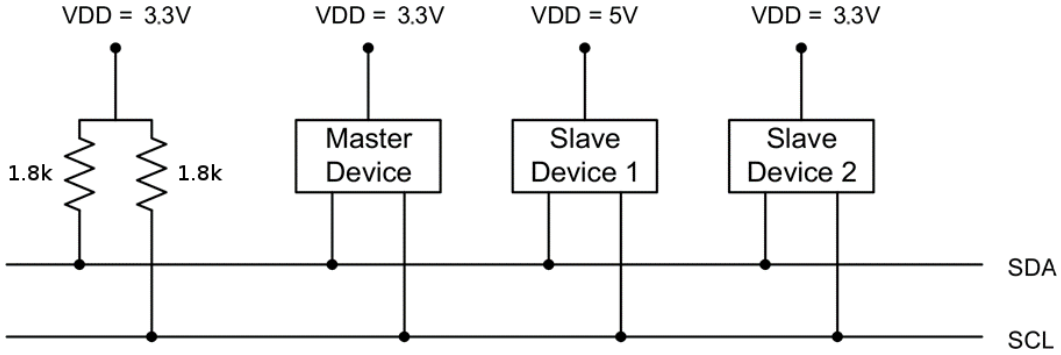

Figure 4.4 - Configuring multiple slaves on a single master

Note: Additional hardware – a run/stop slide switch connected to any unused I/O pin of the team's choice should still be used. We recommend Port 3 pins, as these pins will also have switches on the gondolas. This allows disabling the drive motor on the car so that adjustments can be made with the program running without requiring that the car be placed on a foam block.

Car RF Transceiver Module

The radio frequency (RF) transceiver module on the car communicates with the matching USB RF transceiver module connected to your laptop. This is used to establish a serial connection in the same way the wired serial cable was used, which allows data to be transferred between the car and your laptop. With this, commands from your laptop are sent to the car and output from the car sent to your laptop through your terminal emulator.

The car RF transceiver module requires 5 V power (pin 4, black wire) and ground (pin 2, orange wire) lines. Pin 5 (white wire) is also grounded. To send and receive data, it also requires connections to TX (pin 1, brown wire) and RX (pin 3, green wire), which connect to P0.0 and P0.1 on the EVB respectively. Make sure the 10-pin header is attached to the module as shown in the photo, with the brown wire closest to the side with the 4 jumper pins used to set the baud rate. The wired RS-232/USB and the wireless RF **CANNOT** both be used simultaneously. There will be conflicts. If the wired connection is used the connections to P0.0 and P0.1 on the EVB bus must be temporarily pulled out. If the wireless is used the RS-232 DB9 gray plug must be disconnected from the EVB.

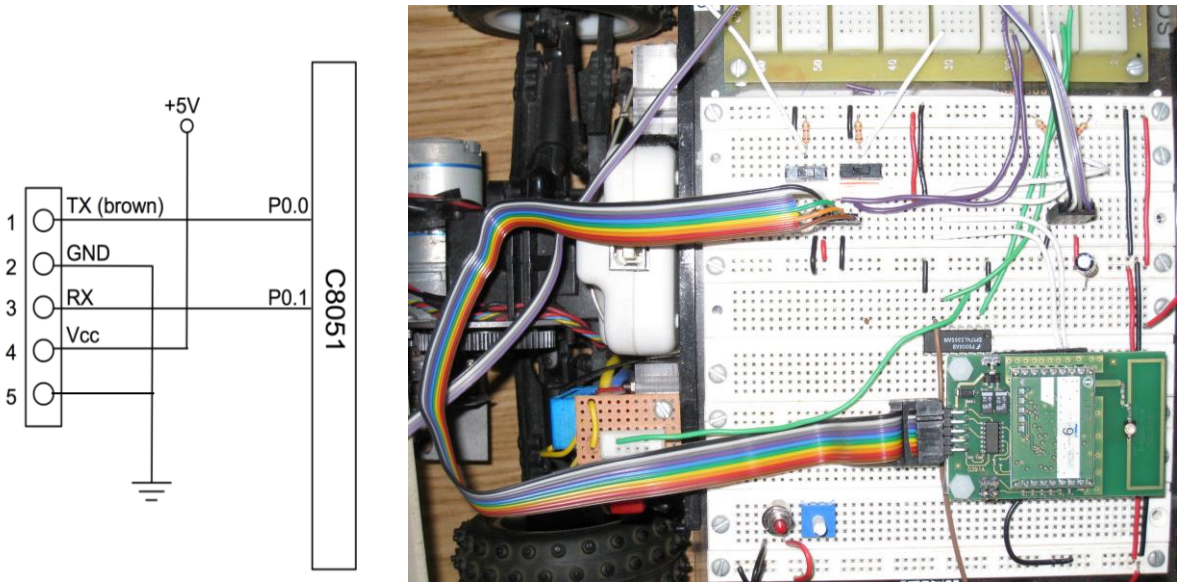

Figure 4.5 – Connections for RF transceiver and photo of module.

Issues/Reminders

There are several other considerations:

1. The steering servo and the speed controller must be updated every 20 ms. The PCA hardware does this automatically as long as it is properly configured.
2. The Compass updates every 33.3 ms, 30 times a second. The car and gondola (Lab 6) controller will work best if each compass reading is a new value, so continue to update the heading every second PCA interrupt, or every 40 ms.
3. The ranger needs 65 ms to complete the ping. Continue to use the PCA ISR to set a flag to read the ranger once every 4th interrupt, or once every 80 ms. In some cases, 100 ms works better. Both the light sensor and range must be read from the ranger after each ping.
5. It is necessary to try several gain constants for the steering gain K_{ps}. The car should attempt to get to the desired heading in a short distance without jerky steering adjustments.
6. What happens if the car goes in reverse? When backing up you should recognize that there will be a problem with the steering control based on the compass and desired heading error unless your controller changes.
7. The condition where maximum error in the heading or ranger corresponds to a maximum pulsewidth should be reevaluated to produce a faster response in the system. This can be accomplished by increasing the gain coefficient, however, care should be taken so that the maximum and minimum allowable pulsewidths are not exceeded.
8. The ranger will be mounted on the top connector so that it is upwards looking. In this position, you can wave your hand above the car to simulate an object being detected.
9. Using `float` variables may cause some unforeseen problems. Integer (`int` or `char`) type variables can represent fractions, for example declare a variable to be `int` in the range from 0 to100 and then do a final divide by 10 in the statement where it is used. As always, to prevent integer truncation, make sure all multiplication operations are performed before any divide operations.

Data Acquisition

When your code is functioning correctly, in addition to drawing a line on the paper, gather data to plot. In order to save the data, you will need to printf() the values to the Terminal screen with the wireless serial link and then copy the output to a plotting utility, such as Excel or MATLAB. Read the terminal emulator section in the **Installing_SiLabs-SDCC-Drivers** manual on LMS for more details. You need to obtain curves for different gains and different directions. You should have ~6 plots. One of those should be what you consider an 'optimal' response, where the car following the field line reasonably well in both the forward and the reverse direction. One or two plots can demonstrate failure cases, where the car lost track of the field line or hit the wall. Make other plots that you consider interesting. Plot the data as a scatter plot or straight-line scatter plot. Make sure the axes show units: seconds or ms on the x-axis, and appropriate units on the y-axis. Read the section **Writing Assignment - Lab 4 Report** (next section) for details of what data to collect for plotting and analysis.

Lab Check-Off: Demonstration and Verification

1. Demonstrate how a desired heading and the neutral drive value are initialized. As stated, the heading must be user selectable. Valid options include a user selectable list.
2. Set the desired heading and gains using the keypad or keyboard. Parameters must be adjustable without recompiling.
3. Place car on floor. Slide switch to run. The steering shouldn't be jerky and should attempt to achieve the desired heading in a short distance of travel. Car will turn in the direction that results in reaching the desired heading the quickest. For example, if desired heading is 0° and the car is started with an actual heading of 350°, it will turn right to achieve the desired heading. If the actual heading is 10°, it will turn left.
4. Use the paper on the classroom floor, placing the car at the starting point. Set the car to head as specified and trace the path the car follows using a felt pen.
5. Your TA may ask you to explain how sections of the C code or circuitry you developed for this exercise work. To do this, you will need to understand the entire system.
6. Display the indicated parameters on the putty terminal.
7. The traced plotted runs must stay completely on the paper when a line is drawn. The rest are allowed to go off as long as they respond correctly to the obstacles and keyboard commands.

Writing Assignment – Lab 4 Report

A brief written memo (see the last 3 pages of this file) plus plots and their analysis is required for this lab (**neither** pseudo-code **nor** C program-listing should be included but the code must be uploaded to LMS, and **NO** formal schematic is required). The memo should have ~6 response plots with different proportional gains and desired headings. `You should adjust the steering gain, adjust the speed up and down. Plot the compass value, direction error and steering pulsewidth on the same time axis. Plot the ranger value, ADC1 value, and drive motor pulsewidth on the same time axis. Values will need to be scaled to plot nicely. Analysis of the plots should explain what is happening and why. Note: **not all plots need to demonstrate 'good driving'**. Explain why certain values prevented the car from meeting the desired results. Significant features on the plots should also be noted and explained. You should say whether a high or low drive speed performed better while controller gains were being optimized.

Finally, the complete commented C source code must be uploaded to LMS.

Grading – Preparations and Check-off

Prior to starting the laboratory, you must complete

1) The appropriate Worksheets (Worksheet #10).
2) The Pin-out form (Port, Interrupt, XBR0, PCA, SMB initializations)
3) The Pseudocode (Revision when finished)

When you are ready to be checked off, the TAs will be looking at the following items:

4) All indicated project requirements are performed correctly (defined above in ***Lab Description & Activities***)
5) Appropriately formatted and commented source code
6) Clean and neat hardware, with appropriate use of colors for source and ground connections

Additionally, you will be asked a number of questions. The questions will cover topics such as

7) Understanding algorithms in the code, identifying locations in the software that perform specific actions, understanding the hardware components, understanding the test equipment

The above 7 items each have an individual contribution to your Laboratory grade.

Lab 4 Report

Group names: _____ _____ _____

The following list is the material that would logically be included in this **brief** report. It is not intended as a direct guide. You have done the experiments, so you have the best idea of what you needed to complete the project and what you learned during the process. This means that the following information may be incomplete based on your experience with the project. If you feel part of what you did during the laboratory is pertinent, include that information.

(suggested # of pages of text (not including plots), but you may go higher)

Introduction	(<1pg)	**20**	
Purpose/Objectives	_____	_____	
Overview of steering & drive control	_____	_____	*Sum:* _____
Results, Plots, Analysis of Plots, & Conclusions	(4-5pg)	**60**	
Verification (how was performance to specifications tested)	_____	_____	
Logical layout of data & Presentation of Plots	_____	_____	
Labeled axes with units	_____	_____	
Analysis of plots from tabulated terminal data w/ explanations	_____	_____	
Problems Encountered & Solutions	_____	_____	
Suggested improvements to HW & SW	_____	_____	*Sum:* _____
Code			
NOTE: no code listing is requested here, but it must be included in the Lab Notebook and you must upload your **Fully Commented** .c file to LMS under **Assignments > Lab 4, only 1 per team**			
Formatting & Neatness		**20**	
Cover Sheet (names, sec/side, grading TA)	_____	_____	
Spelling & Grammar	_____	_____	
Required: Academic Integrity and Division of Labor page - **signed** (See the provided template form)	_____	_____	*Sum:* _____
Lateness			
-20% per School Day	-20 x _____	_____	*Sum:* _____
Total		**100**	***Total Points:***_____

NOTES: **No reports will be accepted if the team has not been checked off for Lab 4. No report grades will be given without uploading softcopies of the .c file to LMS for archival purposes in addition to the signed hardcopy of the report. Use last initial of members in the file name (ex. 2B_HHO_lab4.c for a team in section 2, side B with last names Hamlet, Othello, and Shakespeare). Only one team member should upload the file but it must contain the names of 3 members in the header comments. Everyone on the team must sign the hardcopy of the report using the Academic Integrity form given below.**

Be sure to read "(Bad) LITEC_Report_examp_graphs" under this rubric on LMS to avoid common mistakes when plotting data. Axes must be scaled and units specified. These plots are from Lab 6 data but they are being used to show bad plotting formats (independent of the data).

A **brief** report is required to be turned in after completing Laboratory 4. This is expected to be about a 6-page written document with plots included (~two pages of text). The pseudo-code was required to be submitted to Gradescope and explain how the control was implemented. The report should explain how the plots show the control is working. The results must verify the performance. Not all plots need to work correctly – bad parameter values will lead to unsuccessful runs, which should be documented as such.

The report must include:

1. Analysis of the plots explaining what is happening and why. Significant features on the plots should also be noted and explained. Include a discussion of how the code performs the desired control by adjusting the steering. Indicate points in time when the ranger detection resulted in a change of direction.

2. Collect data for a number of response plots. The values plotted each semester will depend on the particulars of the current objective. Choose your parameters appropriately to validate that the objective was met successfully. For example, you may wish to plot the compass heading reading (-180° to +180°), the ranger value, and (steering PW – center PW) vs. time. Different lines on the same plot can be scaled appropriately to have similar magnitudes. For example, to match the ±180° heading and the steering pulsewidth can be scaled to +/- 100%, as shown below.

$$PW\% = \frac{(\text{PWcount} - \text{PWcenter}) \times 100\%}{(\text{PWmax} - \text{PWmin})/2} = \frac{(\text{PWcount} - \text{PWcenter}) \times 200}{(\text{PWmax} - \text{PWmin})}$$

   Alternatively, the heading may need to be multiplied by 10 to match the range of the motor PW %, but use the same scaling for all plots. Consider a similar scaling when plotting ranger values, A/D results and drive motor pulsewidths. You can also use multiple y-axis on a single plot, though, the association between a line and its axis should be clear.

4. Although not requested for the report, the fully commented C code file must be uploaded on LMS to be given credit for the report.

5. Do not forget to include a cover page with the team member names, section, side, Grading TA name, and report name. Also, you are required to include a division of labor sheet at the end that is signed by all team members.

Academic Integrity Certification *(this part is required exactly as stated)*

All the undersigned hereby acknowledge that all parts of this laboratory exercise and report, other than what was supplied by the course through handouts, code templates and web-based media, have been developed, written, drawn, etc. by the team. The guidelines in the Embedded Control Lab Manual regarding plagiarism and academic integrity have been read, understood, and followed. This applies to all pseudo-code, actual C code, data acquired by the software submitted as part of this report, all plots and tables generated from the data, and any descriptions documenting the work required by the lab procedure. It is understood that any misrepresentations of this policy will result in a failing grade for the course.

Participation *(this is only a template; make changes as appropriate or necessary)*

The following individual members of the team were responsible for (give percentages of involvement)

Hardware implementation: _____
(wiring & pin-out sheet) _____
_____

Software implementation: _____
(pseudo-code & code) _____
_____

Data analysis (if relevant): _____
_____
_____

Report development & editing*: _____
(schematic, diagrams & plots) _____
_____

The following signatures indicate awareness that the above statements are understood and accurate.

_____

_____

_____

*Note, report development/formatting does not constitute an engineering contribution toward successful laboratory completion.