

LAFAYETTE COLLEGE | ELECTRICAL AND COMPUTER ENGINEERING

JGB USER MANUAL

ECE 492 SPRING 2017

Prepared by Raji Birru

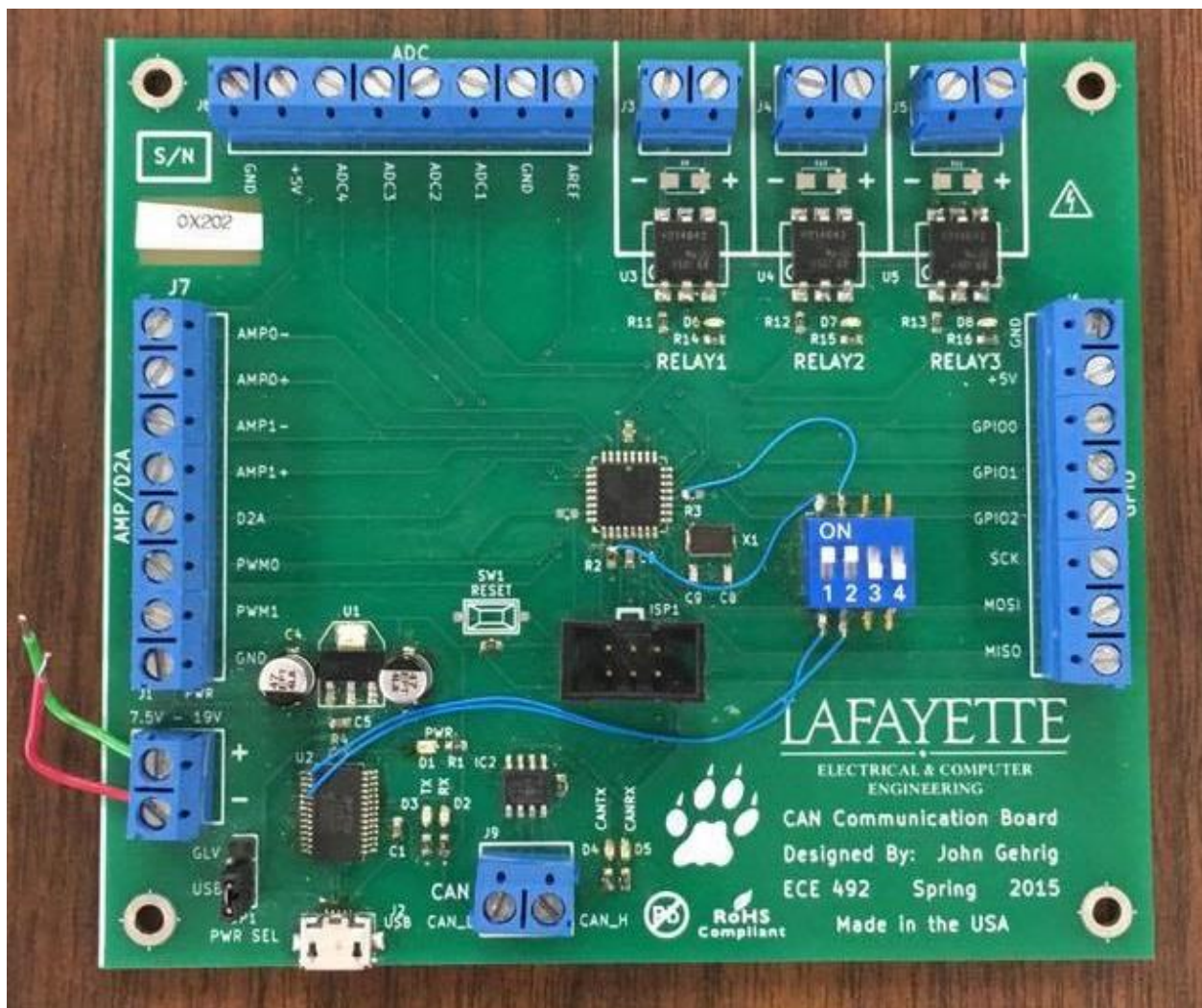
OVERVIEW

This document details the operation and specifications for the JGB (John Gehrig Board) subsystem within the LFEV 2017 project. This document also details all setup operation and configuration settings that are required for the proper operation of the JGB.

The JGB is a CAN bus data acquisition device whose primary functions include but aren't limited to CAN Dump (Printing CAN packets on the line) and Keyboard to CAN (sending user specified data over the CAN Bus). The JGB employs the use of an ATMega16M1 microcontroller for sending receiving over CAN bus and other capabilities which will be discussed in further detail below.

All relevant information about the JGB including a BOM, hardware schematics and Gerber files can be found at the JGB section of the project website. (See here - <http://sites.lafayette.edu/ece492-sp17/subsystems/jgb/>) and are also linked at the end of this document.

Below is a picture of a standard JGB. It will be referenced throughout the document.



HARDWARE

Microcontroller

The JGB employs the use of an ATmega16M1 microcontroller as mentioned above. The microcontroller is in the center of the JGB and can be seen as a small black square. A datasheet has been linked on the project website and it will be instrumental in the development process.

Power

Power can either be provided by USB or a suitable power source. One of these two options can be selected by use of the switch on the bottom left corner of the JGB. If using USB, use a standard micro-USB to USB cable. The USB end can be plugged into any computer while the opposite end can be plugged into the micro-USB port found on the JGB. The appropriate power will be delivered.

Power may also be delivered to the JGB using a power source. If this option is to be used, the power source must be set to 11V, and the aforementioned switch on the bottom left corner of the board must be switched from USB to GLV. The two screw terminals above the switch aptly labeled '+' and '-' are used to provide the power from the source.

Note: Switches 1 and 2 must be set high for proper functionality of the JGB.

Boards

There are currently 3 JGB boards. All should be in the JGB drawer in AEC 400 but this is liable to change. The one pictured above is fully operational. One other board has a bad D2A converter so any tasks that involve that are rendered inoperable on that board. The third has no microcontroller installed. It is therefore recommended that either these boards are fixed or more boards are fabricated. All necessary files for this are located in the JGB repository linked on the project website. The hardware directory in the repo includes Gerber files that can be used to print boards and schematics to populate them.

Debugger

Included with the JGB is an Atmel ICE debugger. This piece of equipment is necessary for programming the JGB. The debugger comes with a proprietary connector and a standard micro-USB to USB cable. The connector is to be plugged into the port labeled ISP1 located right below the microcontroller on one end and the 'AVR' port on the debugger on the other. This step in particular is very important and the SAM port must be avoided.



SOFTWARE

All code relating to the JGB is stored in the repository found on the project website. The repository contains two sections at the moment. Hardware contains all the work related to reproducing JGB hardware. Software contains all code. In the 'src' directory all of John's original files are stored for reference. The directory Project 1 is where all of my work is located. The code in Project 1 is a fork of John Gehrig's work in 'src' and it is recommended that future work continue from there. All work was done in Atmel's proprietary IDE Atmel Studio. Project 1 is an Atmel Studio project and work can be continued there. But the specific classes and headers can be extracted if opting to start a fresh project instead.

UART

All code concerning UART functionality is in 'uart.h' and 'uart.c'.

CAN

All code concerning CAN functionality is in 'can_lib.c', 'can_drv.c' and 'can_frame.c' and their corresponding header files.

PWM

Work on PWM functionality has begun but is very preliminary. The microcontroller onboard the JGB achieves PWM by use of the timer located on it and as such the files 'timer.c' and 'timer.h' handle all PWM functionality.

Configuration

The file 'config.h' is used to configure some default values like default baud rate and buffer size. Any default sizes that may come up in the future should be added here as well for clearer code.

Ring Buffer

The UART code employs use of a ring buffer for managing the data coming in and out. Ring buffers are recommended for use in microcontrollers for its space conserving properties.

Note: The files 'jgb.c' and 'jgb.h' contains some work done by Greg Flynn towards getting the D2A operational that would be very helpful towards any venture in this direction in the future.

The 'main.c' contains a demo of the proper operation of CAN bus sending and proper UART functionality as well. The file is well commented and details on how it is to be executed are written there.