

Good Morning Blinds

ECE 5731 Fall 2020 - 12/2/2020



Group Members

Brian Neumeyer



- Bachelors in Mechanical Engineering
- Currently working at Dataspeed Inc. as a mechanical engineer

Aaron Garofalo



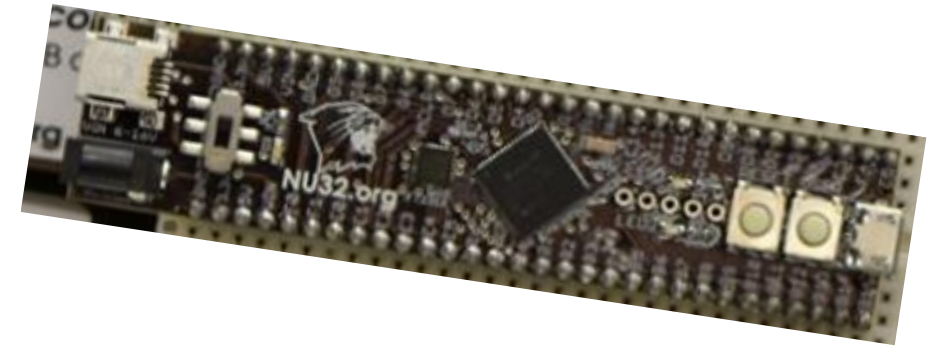
- Bachelors in Mechanical Engineering
- Currently working at MAHLE as a cabin filter development engineer

Lisa Branchick



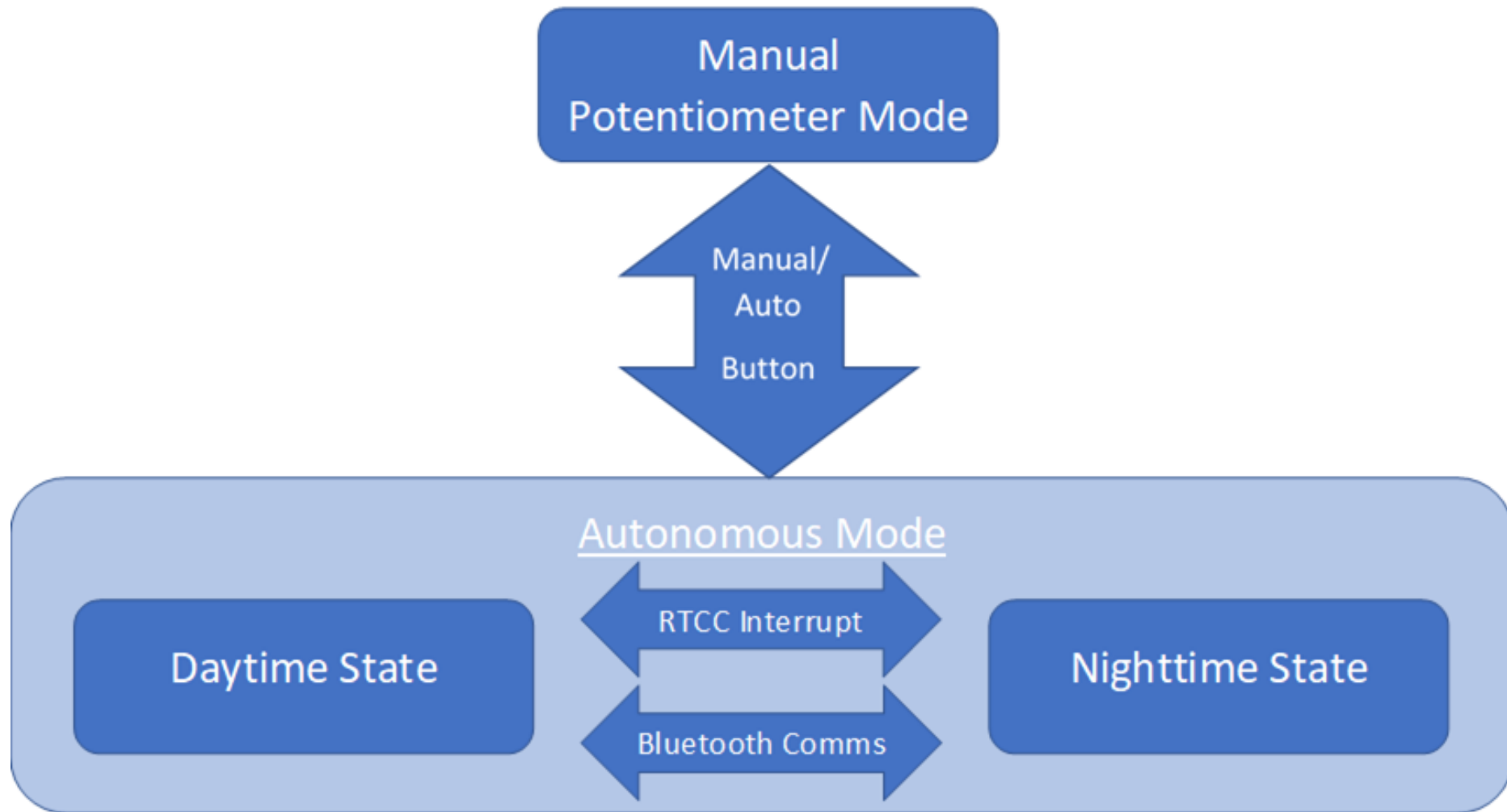
- Bachelors in Mechanical Engineering
- Recently began working at Ford as an embedded controls engineer

Introduction



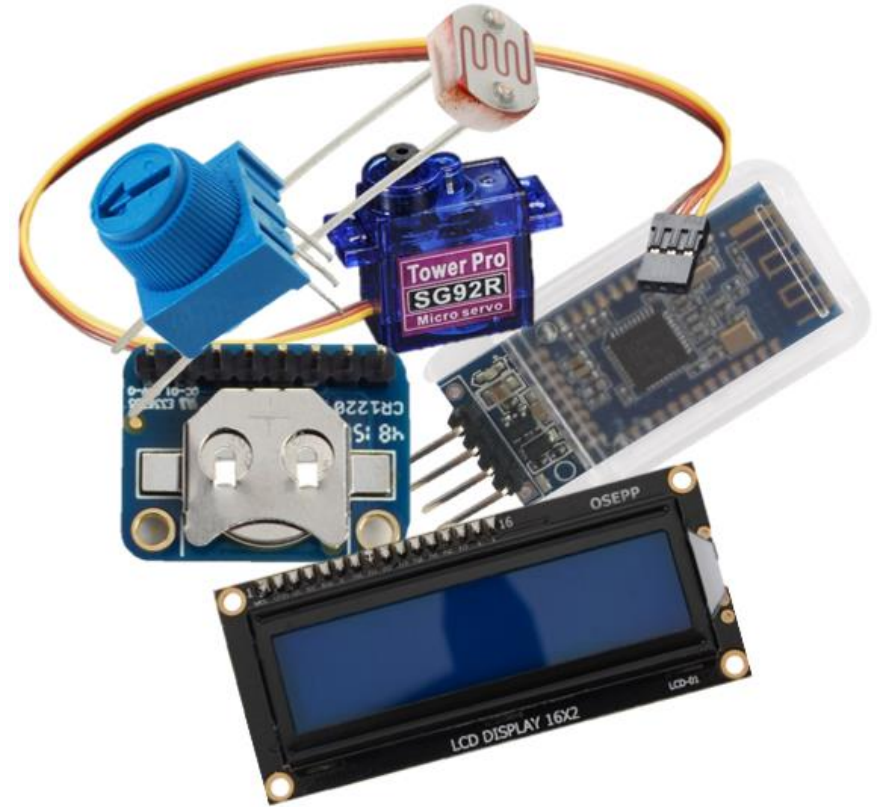
- Good Morning Blinds: Intelligent Window Blind Control
- Project scope: Demonstrate an understanding of embedded controls and concepts using PIC32, actuators, sensors, and I/O devices
- Encompasses bluetooth, timing, interrupts, analog-to-digital conversion, UART, and pulse-width modulation
- PIC32 I/O determines servo position needed to adjust window blinds and convey information to the user via LCD

State Flow Diagram

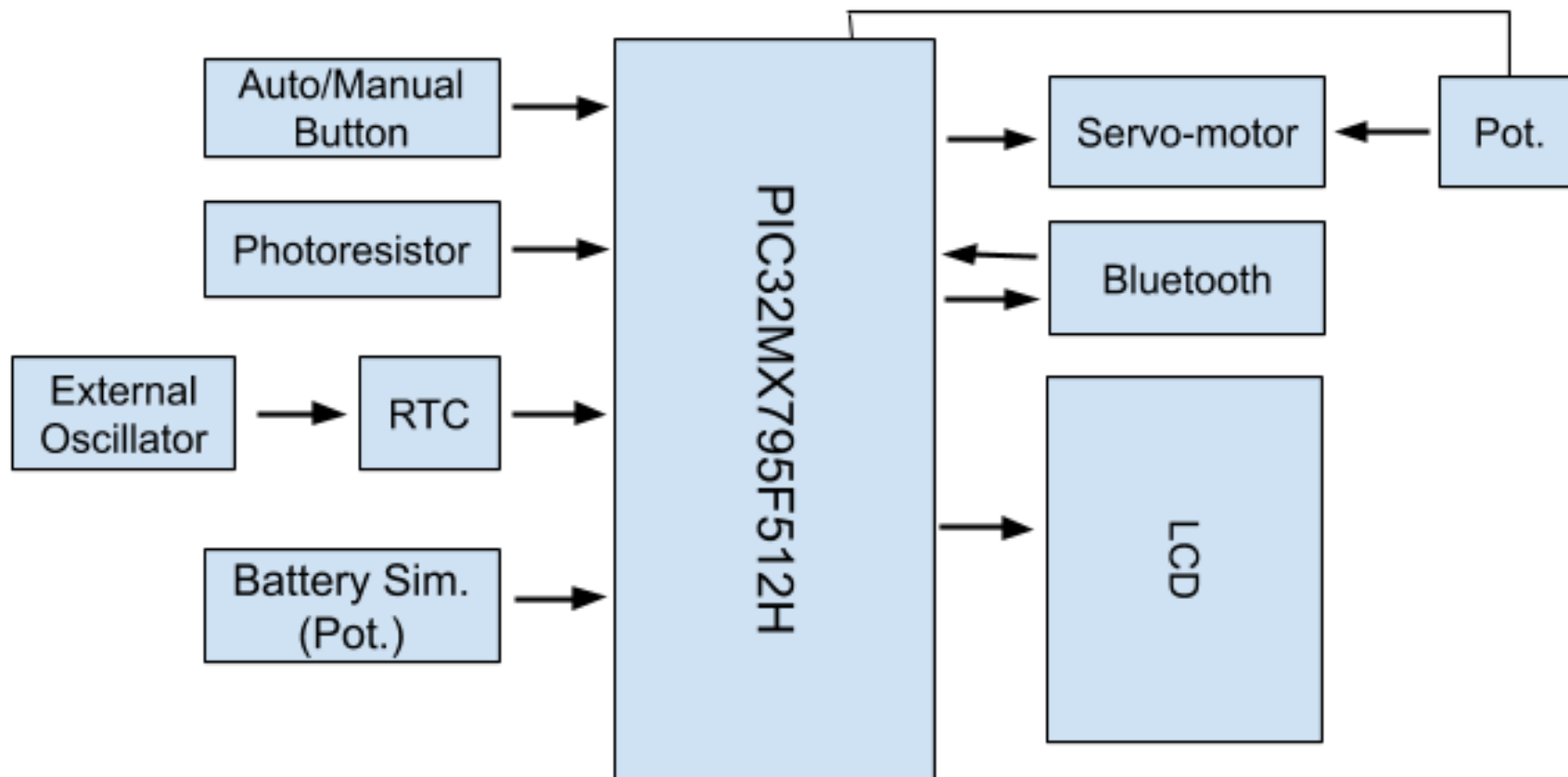


Project Components

- Button: auto/manual mode control
- Potentiometers:
 - Manual adjustment
 - Battery level simulation
- LCD (Liquid Crystal Display):
 - Day/night status
 - Ambient light level
 - Servo position
 - Battery level (simulated)
- RTCC (Real Time Clock and Calendar):
 - Controls automatic mode timing
 - Alternative implementation: timer interrupt in place of RTC
- Bluetooth: system control through Android phone
- Photoresistor: measures light
- Servo: open/close operations



PIC32 Pinout/System Diagram



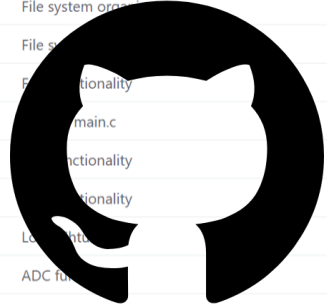
<u>Pin</u>	<u>Function</u>
B12	Manual control potentiometer
B13	LCD RS pin
B14	Photoresistor
B15	Battery Sim. Potentiometer
C13	SOCSO for RTCC
C14	SOSCI for RTCC
D0	Servo PWM
D2	UART RX
D3	UART TX
D4	LCD E Pin
D5	LCD R/W Pin
E0-E7	LCD Pins

Bill of Materials & Cost

<u>Item</u>	<u>Supplier</u>	<u>Price Per Unit</u>	<u>Qty.</u>	<u>Total Per Item</u>
PIC32	Microchip	\$49.00	1	\$49.00
Elegoo Uno Project Kit	Amazon	\$29.99	1	\$29.99
Photoresistor	Elegoo Kit	Included	1	Included
LCD Display	Elegoo Kit	Included	1	Included
Servo	Elegoo Kit	Included	1	Included
Potentiometer	Elegoo Kit	Included	2	Included
Buttons	Elegoo Kit	Included	1	Included
RTC Module	Microcenter	\$4.99	1	\$4.99
Bluetooth	Amazon	\$9.99	1	\$9.99
		Total:		\$93.97

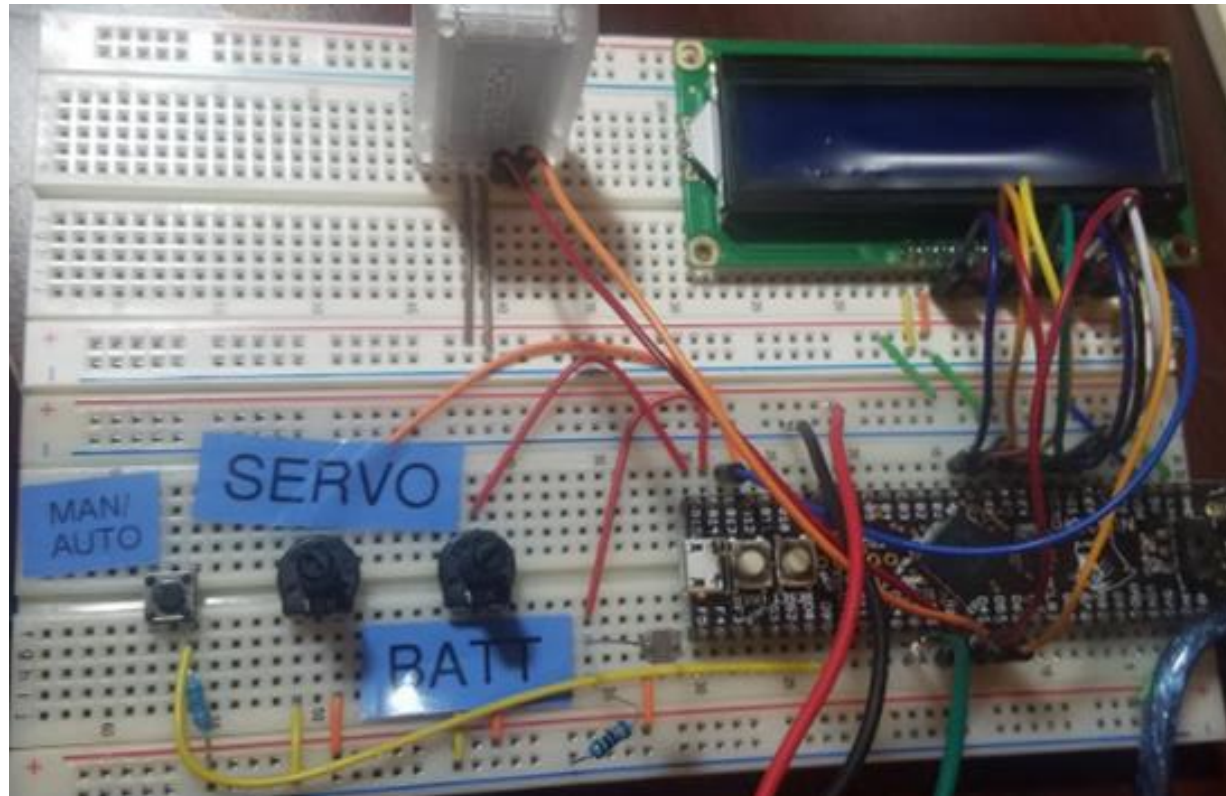
Project Highlights

- Github for collaboration
 - Modular structure to enable parallel workflow
 - Extensive use of supplementary header & c files
 - 650+ lines of code
 - github.com/66hades/ECE5731_Final_Project
- Course topics utilized:
 - Interrupts
 - Digital I/O
 - Timers - 16 & 32 bit
 - PWM (Output Compare)
 - ADC
 - UART → Terminal (w/library)
 - UART → Bluetooth
 - RTCC w/external oscillator
 - LCD



main ECE5731_Final_Project / src /		Go to file	Add file
algarofa Update main.c ...		1 hour ago	History
..			
LCD.c	Modify file structure	8 days ago	
LCD.h	Modify file structure	8 days ago	
Makefile	File system organize	21 days ago	
NU32.c	File system organize	21 days ago	
NU32.h	File system organize	21 days ago	
NU32bootloaded.ld	File system organize	21 days ago	
RTC_Module.c	Full functionality	23 hours ago	
main.c	Full functionality	1 hour ago	
out.dis	Full functionality	23 hours ago	
useADC.c	Full functionality	23 hours ago	
useADC.h	LCD not functioning	yesterday	
useBluetooth.c	ADC functionality	2 days ago	
useBluetooth.h	LCD not functioning	2 days ago	
useDigital.c	Full Functionality	23 hours ago	
useDigital.h	Full Functionality	23 hours ago	
useLCD.c	Full Functionality	23 hours ago	
useLCD.h	Full Functionality	23 hours ago	
useRTCTimer.c	Full Functionality	23 hours ago	
useRTCTimer.h	More Shtuff	yesterday	
useServo.c	useServo functional	5 days ago	
useServo.h	LCD not functioning	2 days ago	

Demonstration

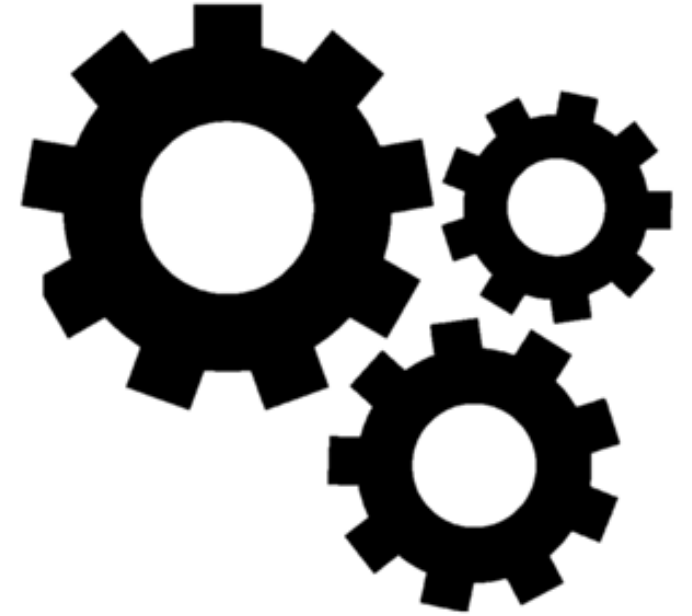


Challenges

- RTCC (Real Time Clock and Calendar):
 - Requires external crystal oscillator
 - Referenced PIC datasheet, reference manual, and online forums
 - Contacted manufacturer - no response
 - Attempted debugging using in-script markers
 - Alternatively explored external module requiring I2C
- Subsystem integration
 - Required multiple .c and .h files
 - Simplified overall code for future debugging
 - Interrupt prioritization



Future Development



- Use on-board RTCC to control timing operations
 - Accurate to ± 0.66 seconds per month
 - User setting for clock time, alarm, and repetition interval
- Interrupt improvements:
 - Control only function flags inside interrupts to improve process flow
 - Analyze priority levels to improve user experience
- Implement battery and solar subsystems
 - Currently simulated for demonstration purposes
- Integrate PID for user-set light level
 - Ambient level is monitored and blind angle is adjusted for consistency

Conclusion

- Produced a functional proof of concept for intelligent window blind control
- Successfully demonstrated all course objectives
- Learned project-enhancing tools GitHub and Visual Studios
 - Allowed for collaboration and productivity





Thank you!



Video

