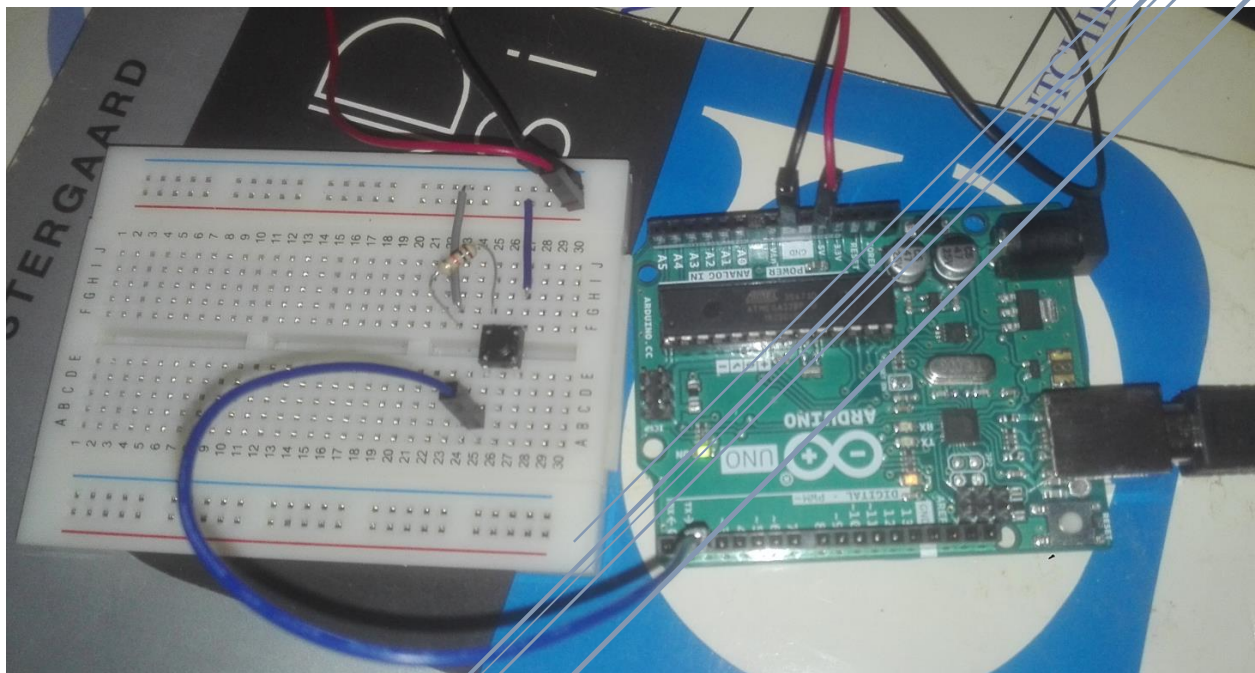


# ELECTIVE C – EXAM PROJECT REPORT

Daniel Blom

Repository <https://github.com/dani153i/kea-embeddedc-exam>



Københavns Erhvervsakademi  
Embedded C 2020

### Which problem I solved

I chose to solve the second assignment (The One-Arm Arduino Bandit), since I had already worked on a Windows-version of this for the second mandatory assignment.

### What I planned to do

I started gathering requirements by defining my systems external input and output.

### External I/O

Actor/Stakeholder	Component/Protocol	Functionality
Player	Push button	Bandit wheels spin when button is pressed.
Technician / Owner	Serial	<b>Menu (input)</b> <ul style="list-style-type: none"><li>• Play Mode</li><li>• Statistics</li></ul> <b>Output</b> <ul style="list-style-type: none"><li>• Menu</li><li>• Bandit information every spin tick<ul style="list-style-type: none"><li>○ Spins available</li><li>○ Bandit wheels</li></ul></li></ul>
Player	Liquid Crystal Display	<ul style="list-style-type: none"><li>• Spins available</li><li>• Bandit wheels</li></ul>

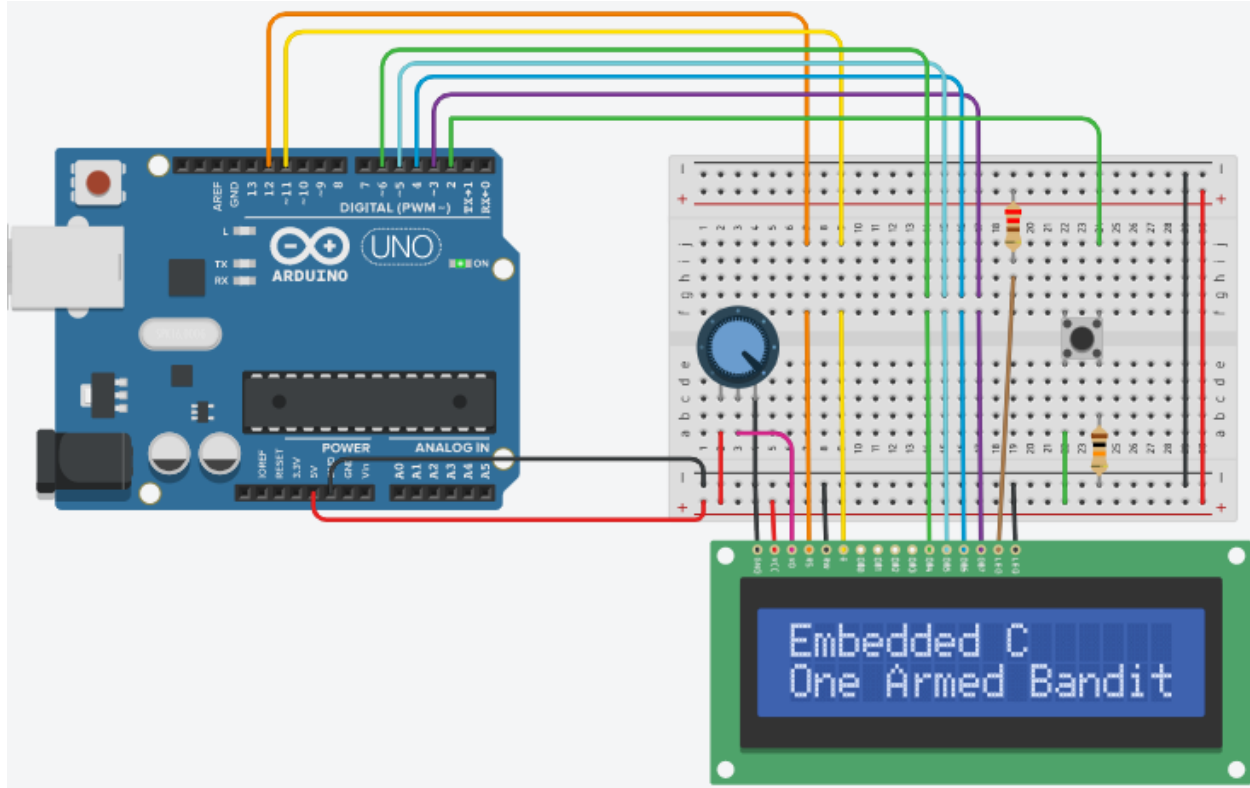
## Circuit Components

I decided on how to make my circuit.

External Input		External Output	
<b>Serial (Arduino IDE)</b>		<b>Serial (Arduino IDE)</b>	
Baud	9600	Baud	9600
Autoscroll	ON	Autoscroll	ON
Input	No line ending	Input	No line ending
Show timestamp	OFF	Show timestamp	OFF

<b>Push-Button</b>		<b>Liquid Crystal Display</b>	
		2 rows, 16 cells.	
<b>Push button</b>	<b>Arduino</b>	<b>LCD</b>	<b>Arduino</b>
D3	D2	D7	D3
D1	+5V	D6	D4
10K Resistor:		D5	D5
Ends to D2		D4	D6
Wiper to GND		enable-pin	D11
		RS-pin	D12
		R/W	GND
		VSS	GND
		VCC	+5V
		10K Resistor:	
		Ends to +5V	
		Wiper to LCD VO pin (pin 3)	



I also planned on implementing some standards.

- Comment my code.
- Write clean, readable, and maintainable code.
- Keep a simple structure to my project.
- Break project source into smaller pieces with specific concerns.
- Separate definitions and implementation in .h and .c files.
- Conserve memory as well as possible.

## What I ended up doing

I used my One-Arm Bandit from the second mandatory assignment, spiced it up and made it run on the Arduino Uno (rev3). The Arduino is connected to a button and can be communicated with through the Serial monitor.

Sadly, I do not have a big enough breadboard for the LCD. Even if I did, it would not look like wheels spinning, since the display only have 2 rows and the player must have displayed the spins available.

## Software Requirement

- Arduino IDE Serial Monitor.  
It must be setup as described in the 'External Output' column under 'Functionality'.  
(I do not recommend use with any other terminal.)

## Functionality

We have two stakeholders who needs to be able to fulfil a few cases each.

Technician/Owner	Player
<ul style="list-style-type: none"><li>• Use Menu<ul style="list-style-type: none"><li>○ Select Play Mode</li><li>○ Select Statistics</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Press button to spin wheels.</li><li>• Use Serial communication to insert tokens. (This should be done with a `Coin/Token Collection Component` instead.)</li></ul>

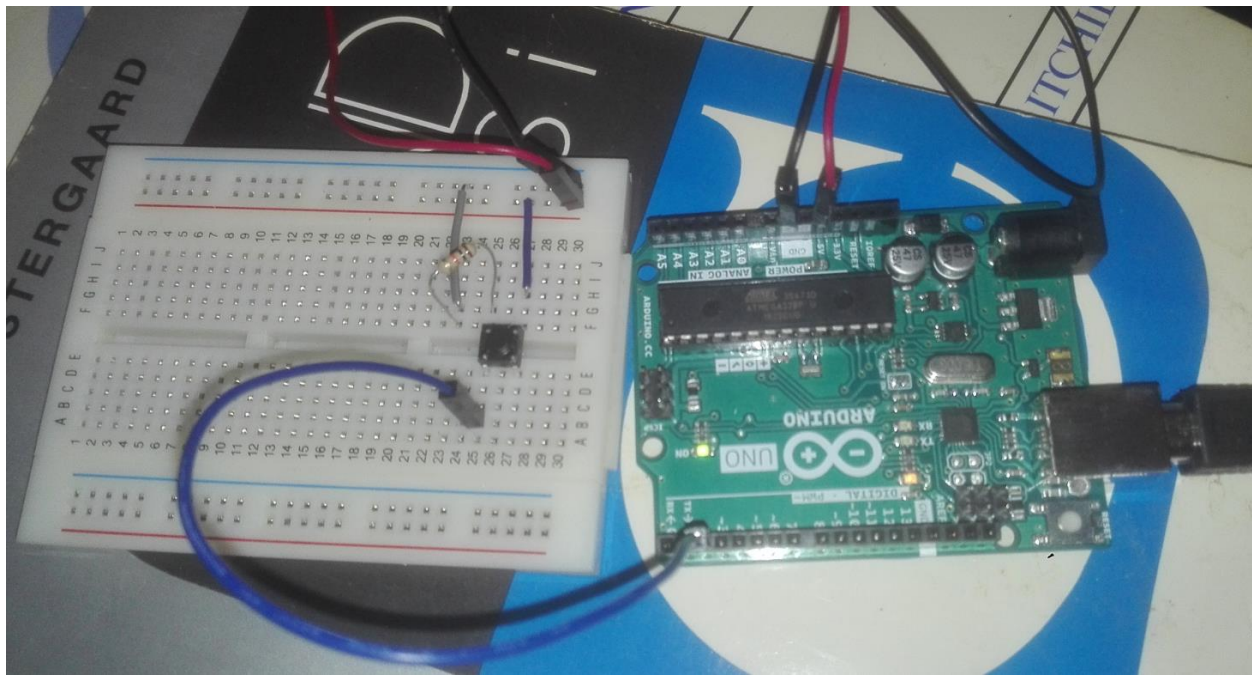
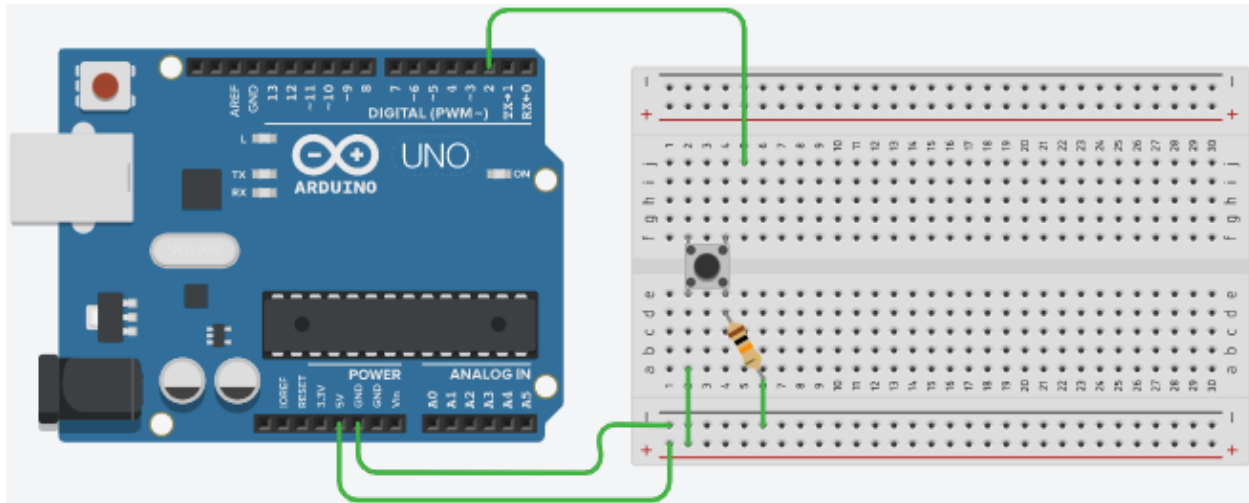
## Circuit Components

Same as intended but without LCD.

External Input		External Output	
<b>Serial (Arduino IDE)</b>		<b>Serial (Arduino IDE)</b>	
Baud	9600	Baud	9600
Autoscroll	ON	Autoscroll	ON
Input	No line ending	Input	No line ending
Show timestamp	OFF	Show timestamp	OFF

Push-Button	
Push button	Arduino
D3	D2
D1	+5V
10K Resistor: Ends to D2 Wiper to GND	



## Reflections

Even though the project source is not as well structured and coherent as I would like, I believe it follows my defined standards quite well considering the amount of time it took to develop.

I had some minor issues with memory allocation, but the only real struggle was implementing the ISR and manipulating registers. I spent a lot of time reading the manual, which is big and quite challenging, that I otherwise could have spent on completing all other tasks. However, the ISR was such a fundamental part of the system, that it had to be prioritized.