WORK PACKAGE 6: ADVANCED PROJECTS

All exercises in this work package should be solved in TinkerCad and using the physical boards.

EXERCISE 1: PARKING ASSISTANCE SYSTEM

You started a small start-up specializing in ultrasonic parking assistance systems for vehicles. A major Swedish truck company contacts you because too many truck drivers cause parking accidents when reversing into a parking spot. Therefore, they want you to design a new parking assistance system.

The new parking assistance system contains an ultrasonic system as a sensor, a multipurpose processing unit, a head-up display with 4 red LEDs, and a speaker.

The requirements for the system are as follows:

- The system shall be able to detect an obstacle behind the vehicle with a maximum distance of 200 cm and a minimum distance of 25 cm.
- The LEDs shall indicate how close an object is to the vehicle. All four LEDs shall be lighted when the obstacle is closer than 30 cm, and the first LED shall light when the object is closer than 200 cm.
- A tone from the speaker shall indicate how close an object is. The tone shall be activated when the object is closer than 250 cm. The closer the object gets to the vehicle, the more "annoying" the tone shall become.

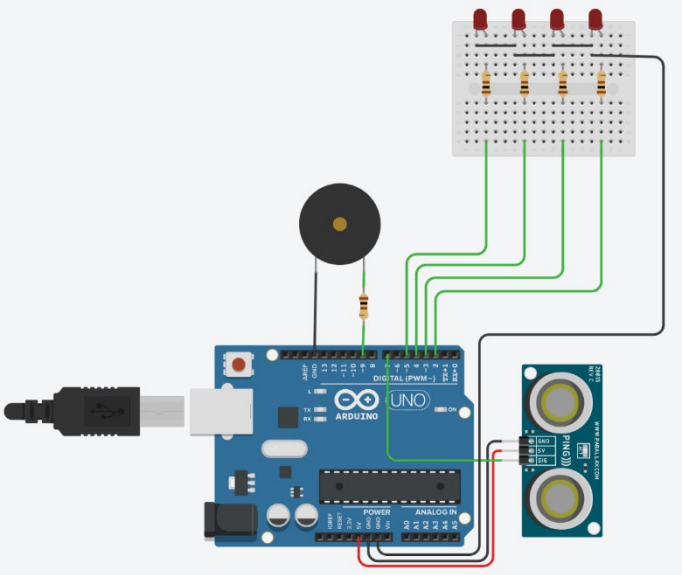

FIGURE 1: PROTOTYPE DESIGN FOR AN ULTRASONIC BASED PARKING ASSISTANCE SYSTEM

- If the object is closer than 25 cm, all four LEDs shall blink, and the tone should become distinctively super annoying to indicate the immediate danger.
- For better maintainability, each feature (i.e., UltraSonic distance detection, LED lights, and Speaker control) shall be controlled through individual functions.

They ask you to prepare a small prototype demonstrating the concept to decide if they want to hire your company for this project. Your "hardware guy" has already prepared a prototype schematic; see Figure 4.

Furthermore, your "hardware guy" reminds you that an ultrasonic sensor works by first sending out a ping, switching to listening, and waiting until an echo returns. She/he says something like this could work:

- Set Pin to Output.
- Write a zero to the output to clear the system. Wait 2 milliseconds.
- Write a one to the output to send a ping. Wait for 10 milliseconds.
- Write a zero to the output to stop sending the ping.
- Set Pin to Input.
- Listen to input if some echo returns. The time until the echo returns is related to the distance of the object from which the ping is reflected.

Your task is to build the system according to the proposed schematic and to develop the necessary software.

TABLE 1: BILL OF MATERIALS FOR EXERCISE 1

Component	Description
Ultrasonic Distance Sensor with 3 PINs[1]	Use the one with 3 pins, not the one with 4 pins. Connect Power and GND to 5V and GND of the Arduino. Connect Sig to IO Pin 7 of the Arduino.
Piezo (Speaker)	Connect to IO/Pin 9 via a 100 Ohm resistance and to GND.
Mini Breadboard	
4x RED LEDs	Connect the 4 LEDs, through 100 Ohm resistors, to IO Pins 2-5 of the Arduino.
5x 100 Ohm resistors	See above
Arduino Uno	
(optional) LCD 2 rows x 16 characters	For exercise 2.

EXERCISE 2: PARKING SENSOR SYSTEM WITH TWO ARDUINO BOARDS

You have probably realized that the setup with one Arduino board for the parking sensor is quite complex and quickly reaches the computation limits. The same processor must handle the ultrasonic distance (echo) sensor, processing the signals and actuating the LEDs.

In this exercise, you must separate the sensing functions from the actuating ones. For this setup, you should use two Arduino boards.

Sensing Arduino: The sensing Arduino should be connected to the Ultrasonic Distance Sensor. It should process the signal and send the distance information to the Actuating Arduino as an integer value.

Actuating Arduino: The actuating Arduino should be connected to the LEDs and Piezo (Speaker). It should receive the distance from the sensing Arduino. Optional: you can use the 2 rows LCD screen to display the distance obtained from the sensing Arduino.

Communication: Arduino boards should communicate using the serial port – pins 0 (Rx) and 1 (Tx).

[1] If you cannot get hold of the 3-pin echo sensor, you can use a 4-pin sensor instead, but you must change the schematics.

EXERCISE 3: STARGATE

You started a small start-up company specialised in software for light effects on film sets. A major Hollywood studio producing a new franchise of the popular sci-fi series “Stargate” contacts you, because they want new cool light effects when the Stargate is dialling out to other planets. To achieve the light effect, the Stargate is equipped with 24 individually addressable RGB LEDs as depicted in Figure 5. A Stargate address contains 7 digits, and only a handful of addresses lead to a successful Stargate connection between planets. A list of correct addresses will be provided.

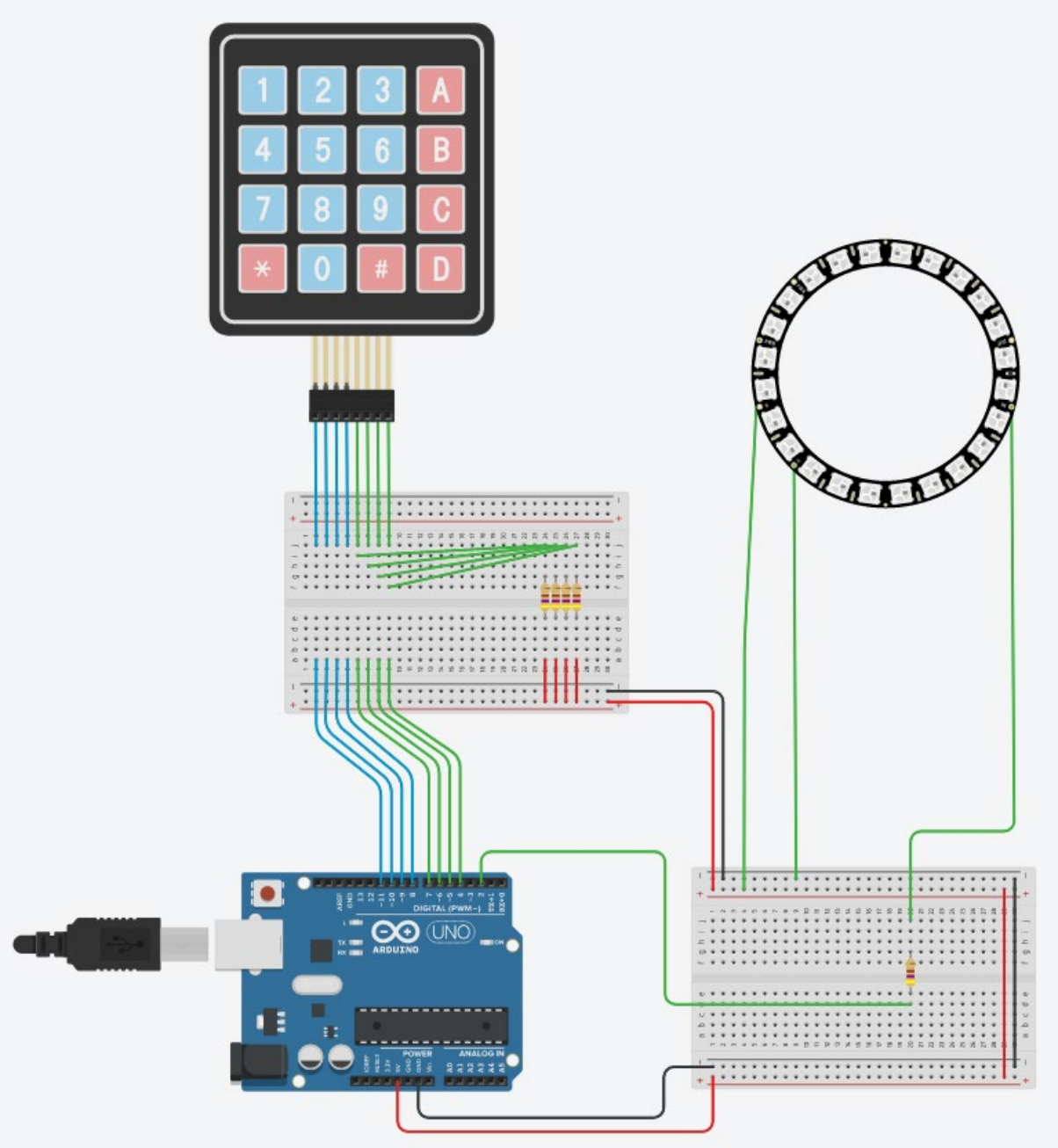

FIGURE 2: SETUP OF LIGHTNING SYSTEM FOR THE STARGATE

Your task is to set up a prototype, as shown in Figure 5. You will be provided with a code template. Complete the template such that the following requirements are fulfilled:

- A Stargate address contains 7 digits. All 7 digits are entered at once.

- As soon as all seven digits are entered, the Stargate starts dialing. Create a fancy light effect that shows that the gate is dialing[2]. For example, let a light spin around the LED ring, change color, or any other fancy animation you can think of.
- After a few seconds (2-3), the first digit "locks into" the Stargate. Locking in means that two LEDs start lighting constantly, indicating that the Stargate accepted that digit (see Figure 6).
- The dialing animation continues for the second digit. Again, after 2-3 seconds, the second digit "locks into" the Stargate. Now, four LEDs at the respective positions light constantly.
- Dialing and locking in continues until 6 digits are locked into the Stargate.
- The seventh digit (two LEDs on top of the Stargate) only locks in if the entered address is in the list of valid addresses.
- If the entered address is invalid, an "error animation" (like all LEDs blinking red or something cool you can think of) occurs instead of locking in the 7th digit.

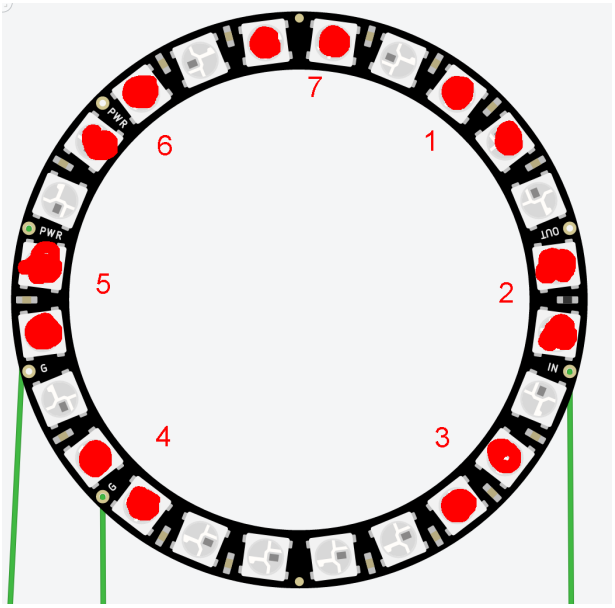

FIGURE 3: LEDS INDICATING SUCCESSFUL "LOCKING IN" OF ADDRESS DIGITS

TABLE 2: BILL OF MATERIALS FOR PROJECT 3

Component	Description
Keypad 4x4	Connected to IO/Pins 4-11 of the Arduino, similar to the previous Work Packages.
NeoPixel Ring 24	Connect, through a 470 Ohm resistance, to Arduino IO port 2. Connect to 5 V power.
Arduino	
5x 470 Ohm resistors	4 are used to prevent bounce back of Keypad, one is used to protect the circuitry of the Neopixel ring.
Arduino Uno	

[2] See from 00:20 https://www.youtube.com/watch?v=24KP9czci8s

EXERCISE 4: PULSE OXIMETER (INSTEAD FOR ONE OF THE EXERCISES 1,2, OR 3)

In this exercise, you will develop a pulse oximeter and visualize the result using the multimeter and the serial port.

You can use the following tutorial for the schematics of such a device: Interfacing MAX30100 Pulse Oximeter Sensor with Arduino (how2electronics.com)

You can borrow the necessary components from the course responsible (I have one sensor) during the course. First come, first served).