ADC LAB:

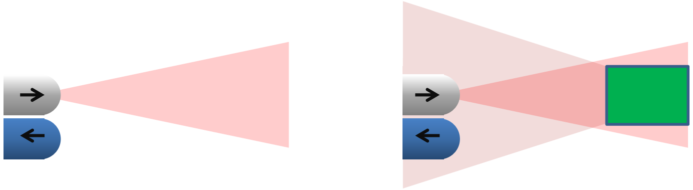
INFRARED PROXIMITY SENSOR

OVERVIEW

For this project you will create a device uses reflected infrared (IR) light to detect if an object is nearby.

DETAILS

The proximity sensor uses an IR emitter (LED) and an IR detector (phototransistor) pointing in the same direction to determine if any object is reflecting IR energy from the LED.

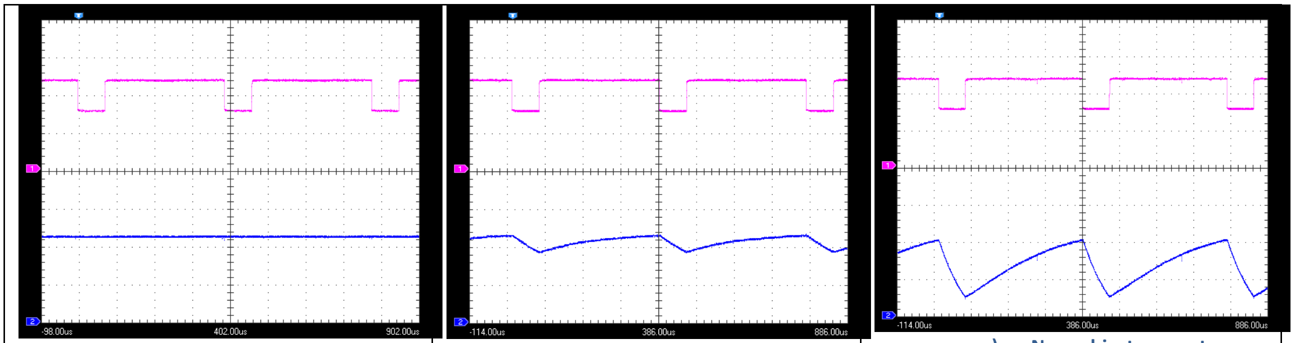


No object present, no
IR reflected back to receiver

Object present, reflects
IR back to receiver

Figure 1. Proximity sensor method of operation.

The proximity sensor works with a combination of hardware and software. Sensing occurs in two steps: First, the software must measure the IR light level (using IR-sensitive phototransistor Q1 and the analog to digital converter) when the IR-emitting LED is **turned off**. Second, the software must measure the IR light level when the IR LED is turned **on**. If the IR brightness level has increased, then there may be an object nearby reflecting the IR from D1 back to Q1. The signal strength is to be indicated by the RGB LED.



a) No object present

b) Far object present

c) Near object present

Figure 2. Oscilloscope screenshots showing IR Output Signal (upper trace) and IR Input Signal (lower trace).

As IR energy increases, this increases the conductivity of the phototransistor and lowers the output voltage. Also, note that the phototransistor has a slow response, as shown by the exponential decay curves in Figures 4b and 4c above.

You can verify that the IR LED is turned on by viewing it with a digital camera (e.g. in a cellphone), as these are sensitive to IR light.

HARDWARE

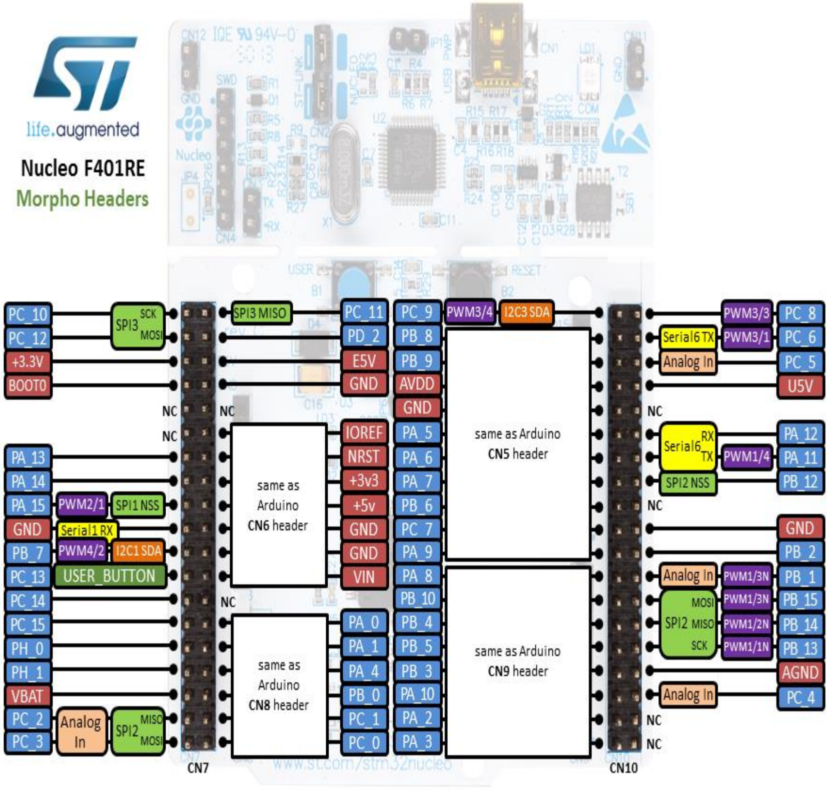


Figure 3. Nucleo-F401RE pinout.

CIRCUIT ASSEMBLY

Build the circuit on the breadboard as shown below.

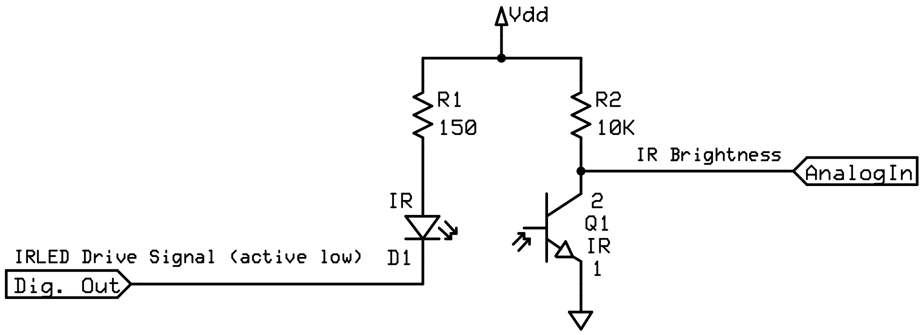


Figure 4. Schematic diagram

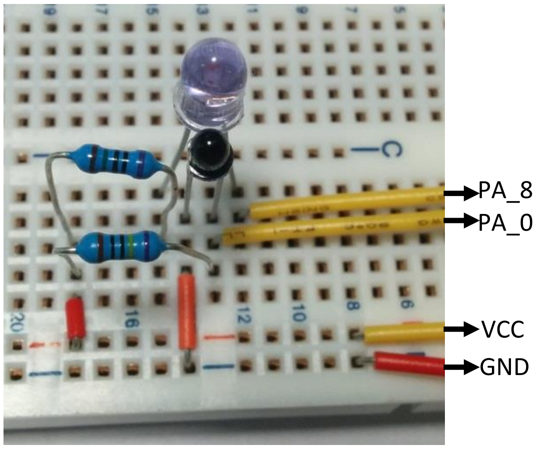


Figure 5. Assembled circuit

Table 1. Components

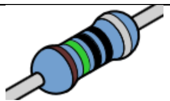
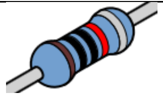
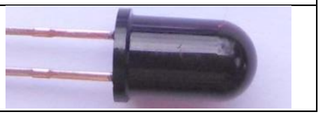
Identifier	Description	Image
R1	150 Ω (brown-green-black-black-silver)	
R2	10 kΩ (brown-black-black-red-silver) Note increasing this will increase sensitivity if needed	
D1	Note that the flat side of the package marks the cathode (negative terminal, long lead)	
Q1	Note that the flat side of the package marks the emitter (negative terminal, short lead)	

Table 2. Signals and connections

Signal Name	Description	Direction	MCU
Dig. Out	Drive signal for IR LED D1	Output	PA_8
Analog In	IR brightness	Input	PA_0
Vdd	3.3 V power supply	Power	
GND	Ground	Power	

SOFTWARE

The control software has the following functions:

- Initialization function: configures GPIO pins and ADC input as needed.
- Delay function: performs delay loop based on function argument.
- Pick LEDs function: Lights RGB LED according to input argument (IR brightness difference).
- Measure IR function: measures difference in brightness caused by lighting LED. In order to reduce noise, average at least twenty measurements before each update of the LED color.
- Main function: Initializes system and then repeatedly calls measure IR, which is passed to the pick LEDs function.

PROCEDURE

BASIC STEPS

1. Load the project into MDK and build it. Download it to the Nucleo-F401RE board in debug mode.
2. Add the following variables to the watch window. Enable periodic window updates in the View menu. Remember that only static variables can be monitored live.
   - reads
   - background_ir
   - differences
   - avg_diff

ANALYSIS

AMBIENT LIGHT ANALYSIS

3. Disconnect the IR emitter. Is the phototransistor reading of IR brightness steady or does it vary? If so, how much and why?
4. Reconnect the IR emitter and monitor the IR difference reading. Shield the phototransistor from IR energy emitted from the side of the LED. Does this change the difference signal strength? If so, by how much and why?

TIME DELAY ANALYSIS

5. The sensitivity of the proximity sensor increases as you wait longer to sample the phototransistor’s voltage after changing the IRLED (see Figure 2 above). What is the impact of doubling the default on-time and off-time? Adjust the constants T_LED_ON and T_LED_OFF.

RANGE ANALYSIS

1. What is the maximum distance at which your sensor can reliably detect a sheet of paper? Your hand? Your thumb? Your arm?
2. Does the ambient light level make a difference? Try shading the receiver from room lighting.

CALIBRATION

3. Calibrate your code so that the RGB LED is lit according to object distance, shown below. Calibrate by adjusting the values in the array called threshold, declared in main.c. To ensure consistency, use the same object for all calibration and testing.
    - Green: No object within 20 cm
    - Blue: 16-20 cm
    - Yellow: 12-16 cm
    - Red: 8-12 cm
    - Magenta: 4-8 cm
    - White: 0-4 cm