

# The ICU-Duino Arduino Shield!

Brought to you by:

**ENGINEERINGSHOCK ELECTRONICS**

## **FEATURES:**

On Board PIR (Passive Infra-red) Motion Sensor

Red Indicator LED

Infra-red (IR) Sensor

Large Prototyping Area with Lots Of 5v & Ground Lines

Each Access to ALL Arduino Uno Pins

On Board Processor Programmed With Several Uses

## **The ICU-Duino Arduino Shield!**

Thank you for purchasing the ICU-Duino Motion Detector Shield! This shield is compatible with the Arduino Uno only. This manual will discuss the schematic diagram of the ICU-Duino, and will speak upon each of its functions. Thank you for making this purchase/pledging to the ICU-Duino kickstarter campaign! We couldn't have done it without you!

## Table of Contents

---

INCLUDED PARTS (PARTS VARY DEPENDING ON PURCHASE):.....	2
THE VIDEO MANUAL & ASSEMBLY VIDEOS:.....	3
THE ACCESSORIES: .....	3
The Wire Connectors:.....	3
The 2-Pin Header Connectors: .....	3
The Infr-ared Remote Control:.....	4
THE ICU-DUINO BLOCK DIAGRAM:.....	5
The Mode Select Button:.....	5
The Prototyping Area: .....	5
The PIR Motion Sensor Connector: .....	6
The Program Select Jumper:.....	6
Input/Output Select:.....	6
The Infra-red Sensor:.....	6
FUNCTIONALITY: .....	6
The PIR Sensor & PIR_OUT Pin:.....	6
The Infra-red Sensor & IR_OUT Pin: .....	7
The Programmed Microcontroller (Modes of Operation):.....	7
Momentary Button Mode: .....	7
Momentary Infra-Red Mode:.....	7
The Difference Between Repeating & Non-Repeating Remotes:.....	8
Latching Button Mode:.....	8
Latching Infra-Red Mode: .....	8
The Arduino Uno Pins:.....	9
THE SCHEMATIC DIAGRAM: .....	9
CODE SAMPLES:.....	10
PIR Motion Detector with Serial Monitor:.....	10
PIR Sensor Communicates with a 1602A LCD – Activate With Remote!.....	10
Using A PIR Sensor, Infrared Remote Control, Relay Board, Resistor-Based Keypad & 1602A LCD As a Security System!.....	10
THANK YOU!.....	11

## **Included Parts (Parts Vary Depending On Purchase):**

1x ICU-Duino Assembly or DIY Electronics Kit (Components Included)

10x Female-Female Wire connectors

1x 2-pin Jumper

1x Infra-Red Remote Control

Bonus Parts Package (Only Available To Those Who Pledged For This Package):

1x Active-Low Relay Board

1x Resistor Based Keypad

1x 1602A LCD

1x 100 Ohm Resistor

1x 4.7k Ohm Resistor

## The Video Manual & Assembly Videos:

I'm a visual person. The below manual gets down to the nitty-gritty, but the video manual really shows you the hands-on approach. I cover all functionality and all hook-up instructions for both versions of the board in this video. Please take the time to watch it. I've also enclosed the assembly videos for those of you who are building from scratch. Part lists are talked about in these videos.

**Video Manual:**

<https://www.youtube.com/watch?v=2CkByD4GU0U>

**Assembly Video:**

<https://www.youtube.com/watch?v=EosEb8gEfWA>

## The Accessories:

### *The Wire Connectors:*

These wire connectors allow for you to make connections between the ICU-Duino and an Arduino Uno. As well, you can use them to connect between your Arduino Uno and other peripherals, such as LCDs, relays, keypads, etc.



### *The 2-Pin Header Connectors:*

You will need one of these bad-boys to choose between program#1 and program#2 by selecting PG1 and PG2 on the program jumper.

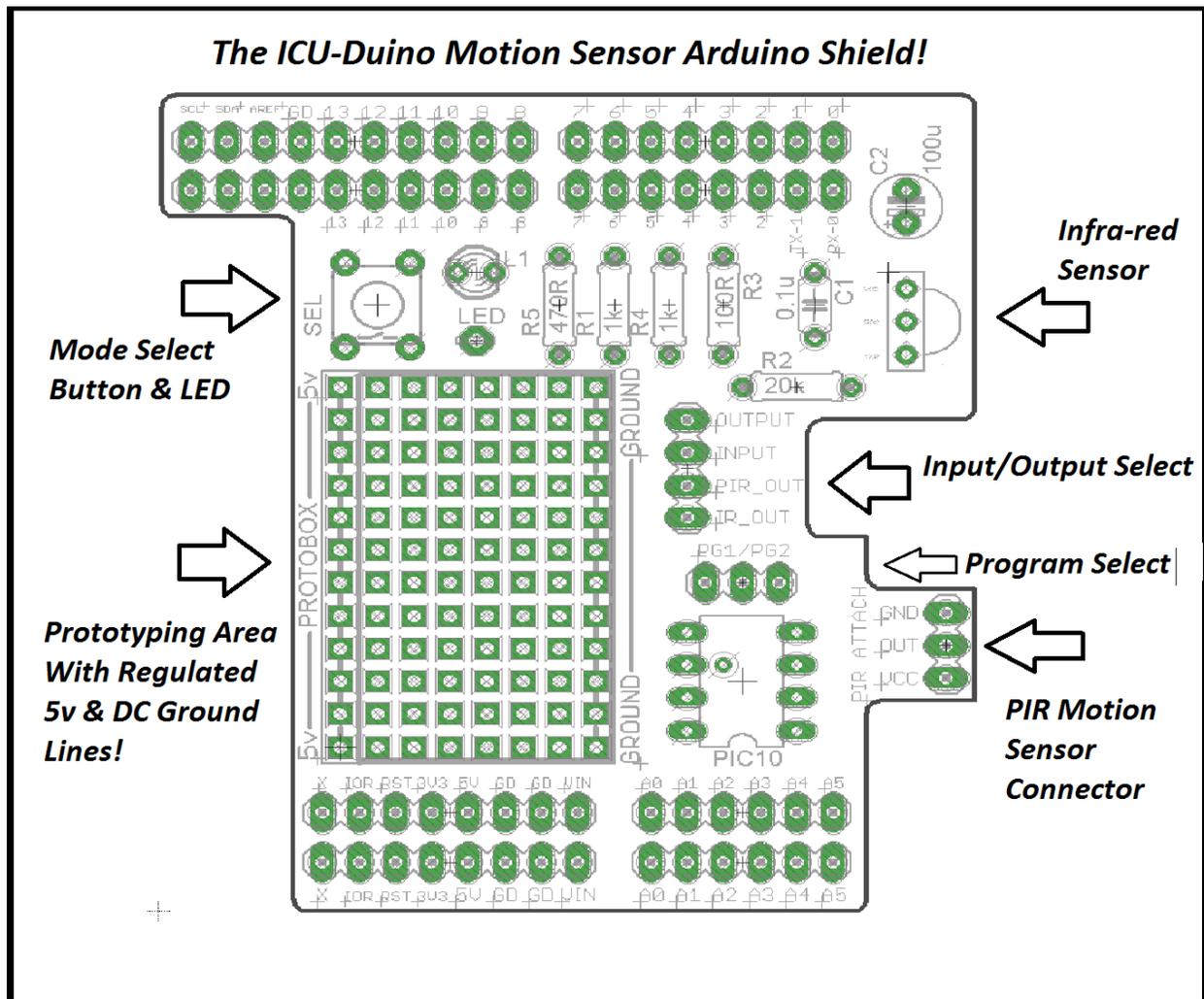


### ***The Infr-ared Remote Control:***

This remote is a simple 38kHz Infra-red non-repeating remote control. It has 21x buttons. You can use your own infra-red remote control, but not all remote controls will be compatible with the ICU-Duino. This specific remote is 100% compatible with this set, and comes with its own battery. In order to apply power to this remote, you will need to remove the plastic tab on the back of the remote. This tab acts as an insulator between the battery and the power terminals on the inside of the remote. Once you remove this tab, your remote will be powered and ready to go. This remote requires a single CR2032 3v battery. You can pick up packs of three from any dollar store for a buck.



## The ICU-Duino Block Diagram:



### ***The Mode Select Button:***

The mode select button “SEL” has two main functions. First of all, each program offered by the microcontroller has two options; button mode, and IR mode. If you power up the ICU-Duino in either of these modes, and you’re holding down the SEL button at the same time, then you will be in either button-momentary mode, or button-latching mode. We will discuss this in the “Functionality” Section of this document. When you are in a button mode, the SEL button acts to keep the OUTPUT pin high for as long as you hold the SEL button down in momentary mode, and to toggle the state of the OUTPUT pin in latching mode. If you find this explanation a little bit confusing, it is all covered in the video manual, and we will be discussing the functions in detail in the “Functionality” section of this document.

### ***The Prototyping Area:***

The prototyping area allows for you to solder in your own circuits. There is a decent amount of space available to you. As well, one side of the prototyping area yields a regulated 5v rail. At the opposite

end, there is a DC ground rail. These rails offer you extra power connections to external devices, such as LCD, relay boards, etc.

### ***The PIR Motion Sensor Connector:***

You won't need to worry much about this area unless you are soldering your ICU-Duino together yourself. This little area is where the PIR sensor is soldered to. The PIR motion sensor module has a power supply pin (VCC), a ground pin (GND), and a signal pin (OUT).

### ***The Program Select Jumper:***

This 3-pin header allows for us to choose between program#1 (PG1 – Momentary Mode), and program#2 (PG2 – Latching Mode). To choose PG1, take your 2-pin jumper and short the middle and left pins together. To choose PG2, take your 2-pin jumper and short the middle and right pins together. We will discuss the modes of operation in the “Functionality” section of this document.

### ***Input/Output Select:***

This 4-pin header is a signal line that allows for you to connect to the output of the PIR sensor, the Infra-red (IR) sensor, the single INPUT of the microcontroller, and the single OUTPUT of the microcontroller. The output of the PIR sensor is labelled “PIR\_OUT”. The output of the IR sensor is labelled “IR\_OUT”. The input pin of the microcontroller is labelled “INPUT”, and the output of the microcontroller is labelled “OUTPUT”.

### ***The Infra-red Sensor:***

This little guy detects signals coming from infra-red remote controls. It is compatible with remotes that offer signals modulated at 38kHz. If you purchased a version of this set that included the Infra-red remote control, then you already have a compatible remote! Take some time to try remote that you have around the house! You might have a few compatible remotes lying around.

## **Functionality:**

The ICU-Duino offers several functions that act to compliment the Arduino Uno. We will be discussing these functions in this section.

### ***The PIR Sensor & PIR\_OUT Pin:***

The passive infra-red (PIR) sensor is a small module with a white fresnel lens connected to it. It is soldered to the board. The PIR sensor detects the infra-red signature coming off of the human body. If you are standing on front of this sensor after it has been calibrated, and you stay very still, then the sensor will “get used to you”. If you move again, it will detect you. As long as the sensor is not blocked by anything, it should be able to detect human presence from all directions. There are two variable resistors on the side of the PIR sensor. These variable resistors are used to calibrate sensitivity.

One of the first things you should do when you power your ICU-Duino up for the first time is to take the output of the PIR sensor, which is the PIR\_OUT pin, and connect it to the LED pin. The LED is a visual

indicator, and is really helpful when calibrating the PIR output. As soon as you power your device on, it is important to know that the output of the PIR sensor is unstable for 10-15 seconds. After this time elapses, it acts as it should. To calibrate, connect the PIR-OUT pin to the LED pin, power the unit up and wait for at least 15 seconds. From there, the LED should light up when it detects a human presence. However, if the LED stays lit up, or never lights up, or simply is acting odd, then you need to play around with the two variable resistors on the underside of the PIR sensor. One variable resistor determines sensitivity while the other determines how long the output will stay high for when motion is detected. If you look at the PIR sensor with the white lens facing up, then the variable resistor on the left is for sensitivity and the other is for the time setting.

The white Fresnel lens acts to filter out ambient light, and protects against false triggers.

### ***The Infra-red Sensor & IR\_OUT Pin:***

The output of the infra-red sensor is normally high, and is only pulsed low when signals are received from a compatible remote control. When a signal is received, the output is pulsed low many, many times a second. To test this, you'll need either an oscilloscope, or a good eye. If you take a female-female wire connector and connect it between the IR\_OUT pin and the LED pin, then the LED will light up. It will flicker quickly if you aim a remote control at the IR sensor and press a button. We will be using the output of the IR sensor in conjunction with our programmed microcontroller, which we will talk about right now.

### ***The Programmed Microcontroller (Modes of Operation):***

The programmed Microcontroller (MCU) has several modes of operation; or rather two modes of operation, and two sub-modes of operation. Confused? Don't be. I'll be explaining in detail. In order to select the required mode of operation, you will need to use the program jumper and the SEL button.

### ***Momentary Button Mode:***

To enter into this mode, you'll need to have the 2-pin header shorting the PG1 pins. You will also need to be holding down the SEL button during power up. After you have the PG1 pins shorted, hold down the SEL button and apply power to your Arduino Uno. You will now be in Momentary button mode. To test this mode, take a female-female wire connection and connect the OUTPUT pin to the LED pin. From here on in, when you press the SEL button, the output of the LED will go from low to high, or rather, from off to on for as long as you hold the button down. As soon as you let go, the LED goes from high to low, or from on to off. This signal is de-bounced. When you press a button, there is always a short time where there is signal bounce, but not in this mode! You can program your code to work around this, but this signal is free of mechanical bounce.

### ***Momentary Infra-Red Mode:***

Before we do anything, take another female-female wire connector and connect the INPUT pin to the IR\_OUT pin. We are now taking the output of the IR receiver and feeding it to the input of the programmed microcontroller. The OUTPUT pin should already be connected to the LED. In this mode,

the SEL button is not used, and will not work for anything. We will be using the remote control to create short single pulses.

To enter into this mode, you'll need to have the 2-pin header shorting the PG1 pins. **DO NOT HOLD DOWN THE SEL BUTTON DURING POWER UP!!!** After you have the PG1 pins shorted, simply power the unit up. As you can see, the button no longer serves any purpose. Take a compatible infra-red remote control and aim it at the IR receiver. If you have a non-repeating remote, simply press a button once, then let go. The LED should turn on, then off shortly after. If you want to keep the output on for a longer duration with a non-repeating remote, then you have to continuously press a button. Any button.

If you have a repeating remote control, and you are in Momentary Infra-red mode, then all you have to do is press and hold down any button. As long as you are holding this button down, the LED will be on. When you let go, it will turn off. This is a single uninterrupted pulse. Please see the video manual if I am causing you some confusion.

### ***The Difference Between Repeating & Non-Repeating Remotes:***

In non-repeating remotes, when you press a button, the remote sends a signal one time. In repeating remotes, when you press a button, the remote sends that specific signal continuously.

### ***Latching Button Mode:***

To enter into this mode, you'll need to have the 2-pin header shorting the PG2 pins. You will also need to be holding down the SEL button during power up. After you have the PG2 pins shorted, hold down the SEL button and apply power to your Arduino Uno. You will now be in latching button mode. To test this mode, take a female-female wire connection and connect the OUTPUT pin to the LED pin. From here on in, when you press the SEL button, the output of the LED will toggle states every time you press the SEL button. For instance, when you first power your unit on in this mode, the LED will be off. If I press the SEL button once, then the output signal (sent to the LED) will turn on, or rather go from 0v to 5v. It will stay in that state until you press the SEL button again. At which time, the output will go from high to low; the LED will turn off. Every time you press the SEL button, the signal will toggle states.

### ***Latching Infra-Red Mode:***

Before we do anything, take another female-female wire connector and connect the INPUT pin to the IR\_OUT pin. We are now taking the output of the IR receiver and feeding it to the input of the programmed microcontroller. The OUTPUT pin should already be connected to the LED. In this mode, the SEL button is not used, and will not work for anything. We will be using the remote control to toggle states. This mode of operation is very similar to the latching button mode, only we use the IR remote control to toggle states.

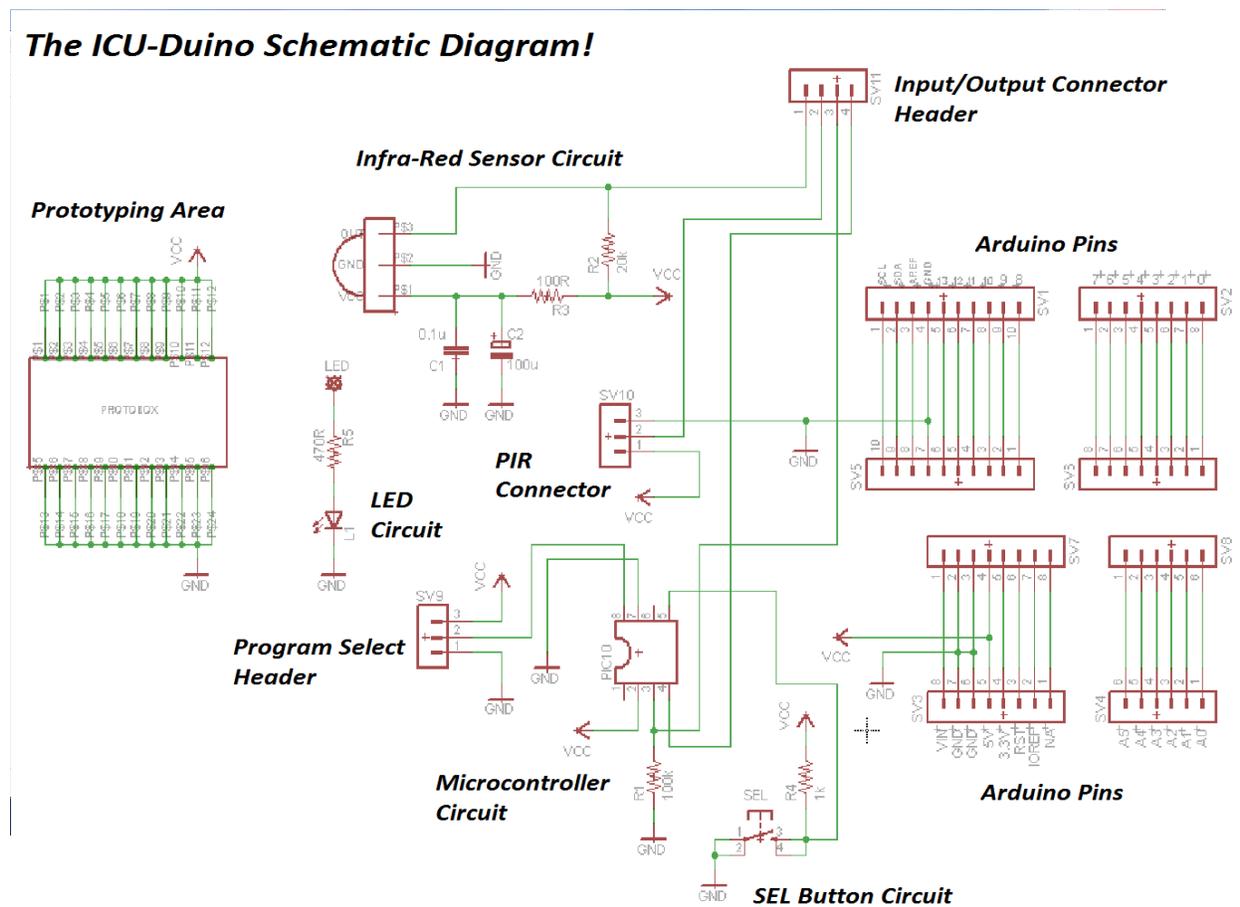
To enter into this mode, you'll need to have the 2-pin header shorting the PG2 pins. **DO NOT HOLD DOWN THE SEL BUTTON DURING POWER UP!!!** After you have the PG2 pins shorted, simply power the unit up. As you can see, the button no longer serves any purpose. Take a compatible infra-red remote

control and aim it at the IR receiver. From here on in, every time you press a button on your compatible IR remote control, the output will toggle states. Just like in latching button mode; only you HAVE to use the infra-red remote control. Not the SEL button.

### The Arduino Uno Pins:

All of the Arduino Uno pins are ALL still available and are NOT connected by default to any of the circuits on the ICU-Duino board. That is, with exception to the 5v and ground lines, which act to power the ICU-Duino shield. It is up to you to implement the functions of this shield. You also have a small prototyping area, which we suggest you find a use for =D

## The Schematic Diagram:



PDF Download of Schematic:



## Code Samples:

In this section, I'm offering up a few pieces of sample code that relate to the ICU-Duino. I hope that you find them helpful. I also have videos to compliment the code.

### ***PIR Motion Detector with Serial Monitor:***

In this video, I'll show you how to count how many times motion has been detected in your absence. Every time motion is detected, it will tell you on the Arduino Serial Monitor!

**Youtube Video:** <https://www.youtube.com/watch?v=czuXv5mFWio>

**Arduino Code:**



IDU\_Duino1.ino

### ***PIR Sensor Communicates with a 1602A LCD – Activate With Remote!***

In this video, you'll use your remote control to activate the program. From there, the LCD will tell you how many times motion has been detected. You need to have a 1602A LCD for this project. Many of you have LCDs coming to you as rewards. If not, they can be found on ebay for decent prices.

**Youtube Video:** <https://www.youtube.com/watch?v=8pEQGNr9PEg>

**Arduino Code:**



ICU\_LCD.ino

### ***Using A PIR Sensor, Infrared Remote Control, Relay Board, Resistor-Based Keypad & 1602A LCD As a Security System!***

This project requires the extra parts bundle offered as a reward for larger kickstarter pledges. If you do not have the relay board, the 1602A LCD, and the resistor based keypad, then this project will not apply directly to you. I apologize for the somewhat convoluted connection video. I hope that you gain enough information to make your connections with relative ease!

**Connection Video:** <https://www.youtube.com/watch?v=f5eegQ4NtWg>

**Demo Video:** <https://www.youtube.com/watch?v=YZ6qbV4nmwM>

**Arduino Code:**



PIR\_ADC\_KEYPAD1.in

o

## **Thank You!**

Many thanks to all of you who pledged towards this kickstarter campaign! I couldn't have done it without you, and I'm sincerely grateful. If any of you have any questions at all, please do not hesitate to contact me directly at [engineeringshock@hotmail.com](mailto:engineeringshock@hotmail.com)

Warm regards, and many thanks,

Pat