



7 Blockly - Dobot to Dobot Handshaking

NAME: _____ Date: _____ Section: _____

INTRODUCTION

Robotic arms need to communicate with other robots in a work cell, or factory. This is called **HANDSHAKING** and can be done between different machines, devices and robots. It is a very simple form of communication and is done with simple ones and zeros; or “ons” and “offs”.

In this activity you will learn how to make a robot handshake with another robot.

Robot 1 will pass a part into another robot’s work envelope, go to a safe position and then send a signal to other robot. The signal received by the other robot will initiate a sequence to get the part, and place it somewhere else.



Caution: NEVER wire anything to the Dobot Magician while it has power on. ALWAYS shutdown the Dobot before making connections or damage to the robot could occur.

KEY VOCABULARY

- Input
- Output
- Function / Voids
- Handshaking

EQUIPMENT & SUPPLIES

- 2 Robot Magicians
- 1” cylinders or cubes
- Suction Cup Gripper
- DobotStudio software
- [Dobot Input/Output Guide](#)
- Handshake Modules



ESSENTIAL QUESTIONS

Essential questions answered in this activity include:

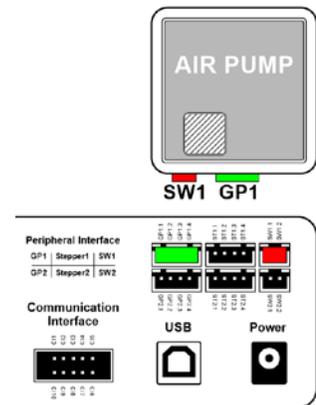
- How do I make a robot send a signal?
- How do I get a robot to receive a signal?
- How is this done in Dobot Studio Software?
- How do I make two robots talk to one another?

PROCEDURE



Caution: NEVER wire anything to the Dobot Magician while it has power on. ALWAYS turn it off before making connections or damage to the robot could occur. Be sure to ask your instructor if you have any questions.

1. Set up both **Robots** with a suction cups - **GP1 & SW1** and place Dobot field diagrams, taped to the work surface, between the two robots.
2. Wire **Robot1** with an **OUTPUT** signal **GP2 - EIO13**.
3. Wire **Robot2** with an **INPUT** signal **GP5 - EIO5**.
4. Wire both Robots to the Handshaking module as shown in the following pages. Be sure that wires are not going to be pulled out by the motion of the robots.



Open Loop System Block Diagram: Robot #1 acts as the input for robot #2. Robot #2 does not communicate back to robot #1 so there is no feedback. The Handshake Module acts as the device that helps make the handshake happen safely.

Order of operations

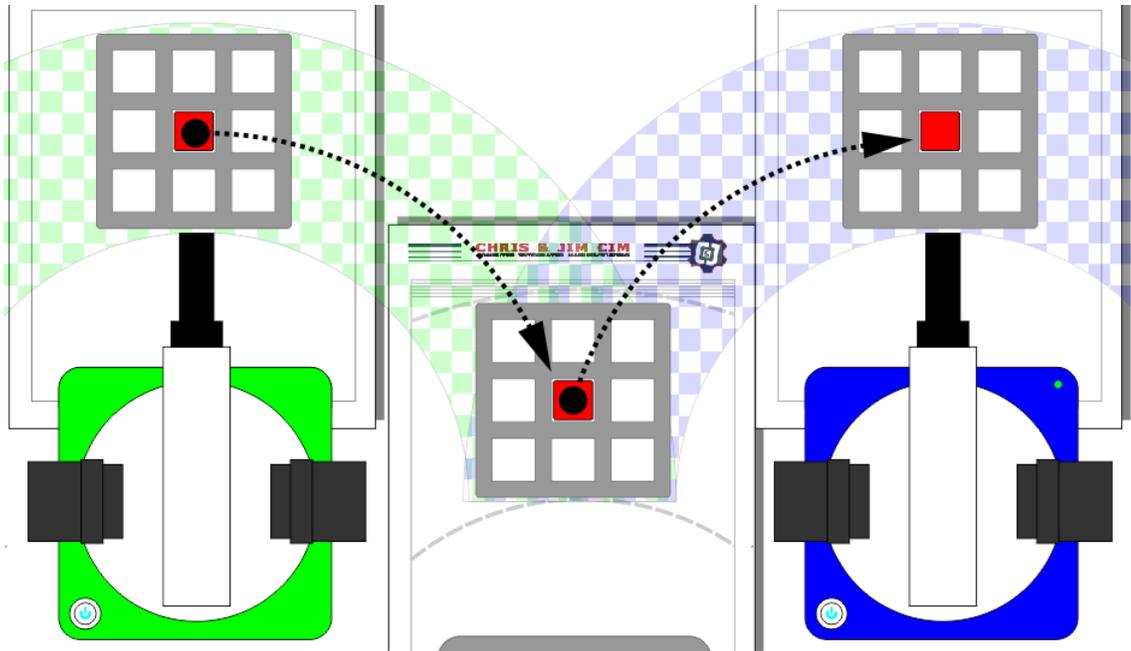
ROBOT 1 - OUTPUT SIGNAL

- Move - Home
- Move - ABPick
- Move - ATPick
- Jump - ATPlace
- Move - ABPlace
- Move - Home
- Send Output Signal

ROBOT 2 - INPUT SIGNAL

- Move - Home
- Wait for INPUT Signal
- Move - ABPick
- Move - ATPick
- Jump - ATPlace
- Move - ABPlace
- Move - Home





Open up Blockly in the software



When you re-open this program check that the name of the file on top matches the code in the file, if it does not, you may end up overwriting another program

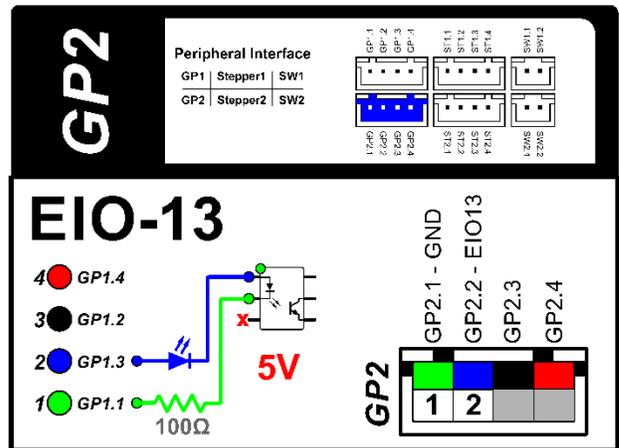
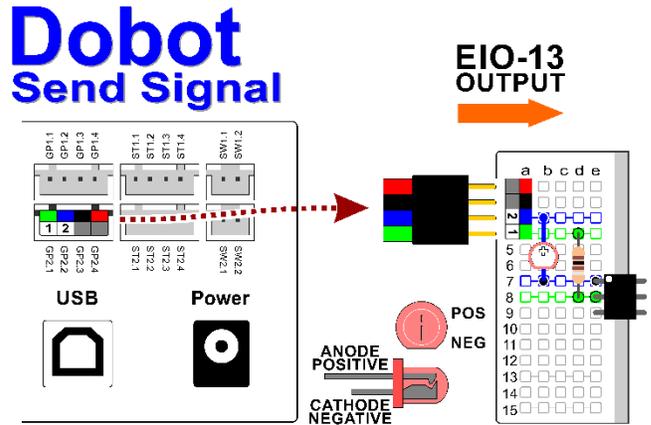


How to SEND an OUTPUT signal.

The first part of this activity will focus on how to send a signal.

In order to make sure our setup, wiring, and program are all correct, we will only use **ONE ROBOT**.

Nothing should be connected to the Output side of the handshake module.



Drag over the **SetIOMultiplexing** block from the DobotAPI/ I/O Toolbox



Change the settings to **OUTPUT 5V** and **EIO13**

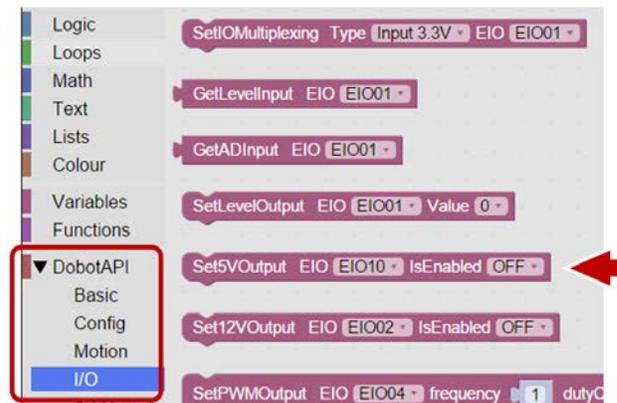
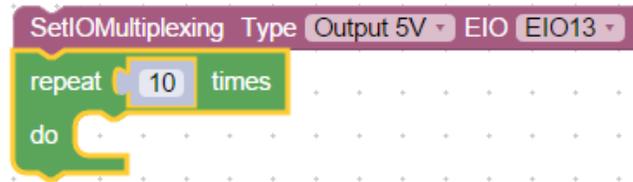


In order to test the OUTPUT signal, we will create a simple program that loops the process of turning on and off the output.

Drag over the **RepeatTimes** Loop

10 times should be enough to ensure our setup is working correctly so make it repeat this many times.

Drag over the **Set5VOutput** block from the DobotAPI/ I/O Toolbox.

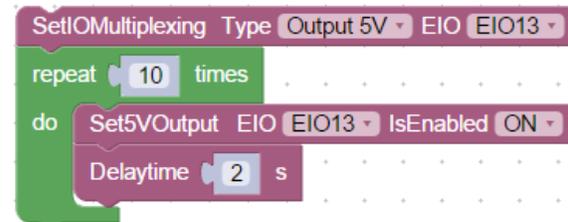
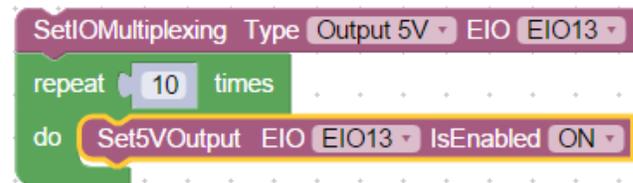


Add the block to our loop

Set the port to **EIO13**

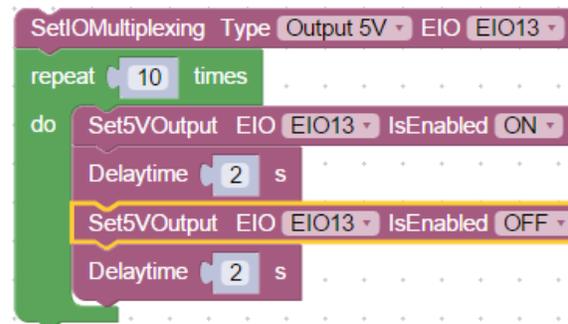
Set Enabled to **ON**

Insert a **TimeDelay** under the **Set5VOutput** block. Set it to send the signal for two seconds.

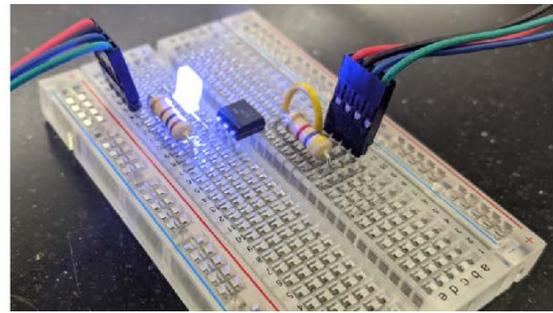
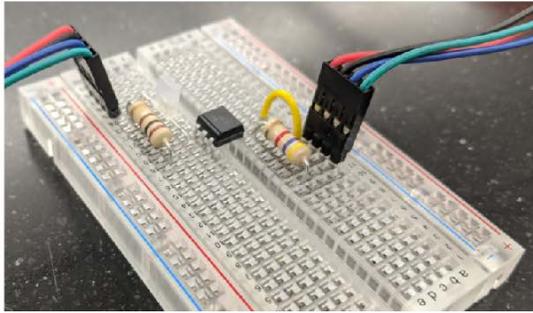


Duplicate both steps

Set Enabled to **OFF**



Once this portion of the program is completed, run it and see if it works correctly. The LED on the handshake module should light up whenever EIO13 is enabled ON. If it does not work, troubleshoot it until it does.



If your set up did not work correctly the first time, what did you have to do to make it work?

TROUBLESHOOTING THE OPTICAL ISOLATOR



The LED helps us determine if the signal is actually being sent by the first robot. We can write a simple Blockly program to see if the second robot is getting the signal. The issue is if the second robot is not getting the signal..... WHY is it not getting the signal? It could be a damaged optical isolator.

An easy way to check is to use a voltage meter set to check RESISTANCE. When no signal is present, the voltmeter should read "O.L" or "---". When a signal is present it should read near the resistance value of the resistor used.

POS

NEG

ANODE POSITIVE

CATHODE NEGATIVE

	a	b	c	d	e	f	g	h	i	j	
1											1
2											2
3											3
4											4
5											5
6											6
7											7
8											8
9											9
10											10
11											11
12											12
13											13
14											14
15											15

Ω

Optical Isolator OFF = 0.L or ---
 Optical Isolator ON = 4.9+ k Ω



Programming for ROBOT 1

Throughout the remainder of this activity, we will use many of the skills that you have developed in previous activities. If you need clarification at any point, you can go back to the other activities for more information.

As we have done in the past,

- Create *Functions*
- Create *Variables*
- Find *Positions*

```
to variables
  set Home X to 999
  set Home Y to 999
  set Home Z to 999
  set Pick X to 999
  set Pick Y to 999
  set Pick Z Above to 999
  set Pick Z At to 999
  set Place X to 999
  set Place Y to 999
  set Place Z Above to 999
  set Place Z At to 999
```

```
to pickandplace
  MoveTo X Pick X Y Pick Y Z Pick Z Above
  MoveTo X Pick X Y Pick Y Z Pick Z At
  SuctionCup ON
  Delaytime 3 s
  JumpTo X Place X Y Place Y Z Place Z At
  SuctionCup OFF
  Delaytime 2 s
  JumpTo X Home X Y Home Y Z Home Z
```

Create a new *Function* - SendSignal

```
to sendsignal
  Set5VOutput EIO EIO13 IsEnabled ON
  Delaytime 2 s
  Set5VOutput EIO EIO13 IsEnabled OFF
```

Put the program together!

```
SetOMultiplexing Type Output 5V EIO EIO13
variables
repeat 10 times
do
  pickandplace
  Delaytime 0.5 s
  sendsignal
  Delaytime 4 s
```



Programming for ROBOT 2

Throughout the remainder of this activity, we will make assumptions that you have already learned several concepts in previous activities

As we have done in the past,

- Create *Functions*
- Create *Variables*
- Find *Positions*

Create a **RepeatUntil** Loop to wait for the signal



Reminder!! Your repeat until will need to wait for a Zero/Low signal for the Magician V2 and a One/High signal for the Magician V1

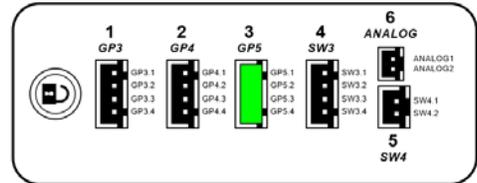
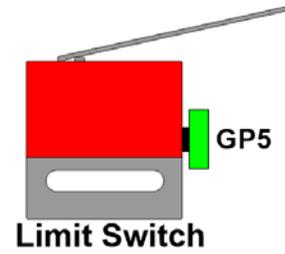
Put the different parts of the program together.



USE A LIMIT SWITCH HARDWIRED TO THE ARM FOR TESTING THIS PROGRAM

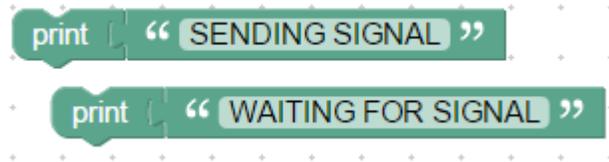
Once this portion of the program is completed, run it and see if it works correctly. If it does not work, troubleshoot it until it does.

When you press the switch, what happens?



If your set up did not work correctly the first time, what did you have to do to make it work?

Consider adding Print to Running Log commands to both programs so that you can see what the robot is doing at any given time.



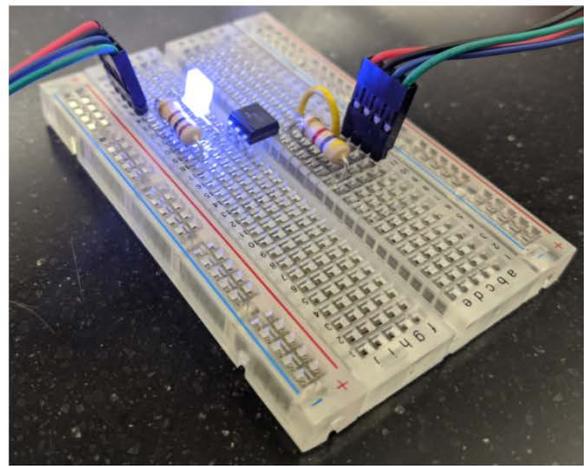
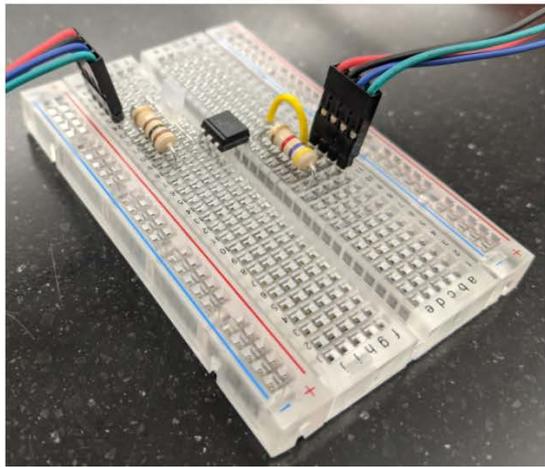
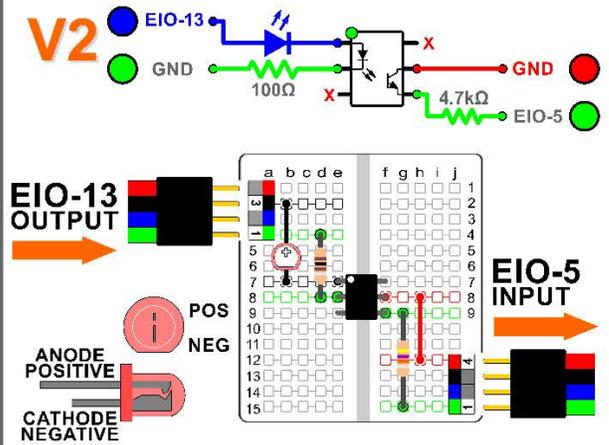
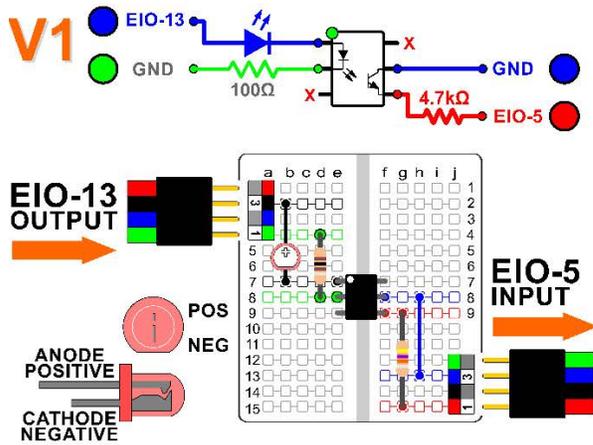
Now that both programs are created **AND** tested, it is time to put it all together.

Finish the Handshake Module and replace the manual limit switch on **ROBOT 2** with the breadboard connection.

Dobot Magician V1 (All White IO Ports)

Dobot Magician V2 (Colored IO Ports)





Be sure to consult the Dobot Input/Output Guide if you want to use other inputs and outputs, as damage to your robot or your other equipment may result.

Once the program is completed, run it and see if it works correctly. If it does not work, troubleshoot it until it does.

If your set up did not work correctly the first time, what did you have to do to make it work?

CONCLUSION



1. *What would you have to do to make this program run five times without any human intervention? Explain fully below.*
2. *What other inputs could you use on your robot to start this process? Use the [Dobot Input/Output Guide](#) to answer this question, and do not attempt to try it without your instructor's permission.*
3. *What other outputs could you use on your robot to start this process? Use the [Dobot Input/Output Guide](#) to answer this question, and do not attempt to try it without your instructor's permission.*

GOING BEYOND

Finished early? Try some of the actions below. When finished, show your instructor and have them initial on the line.

1. Change the LEDs to Motors. Be sure to get your instructor's permission, and be sure to use the correct outputs for the motor chosen.



Be sure to consult the [Dobot Input/Output Guide](#) if you want to use other inputs and outputs, as damage to your robot or your other equipment may result.

