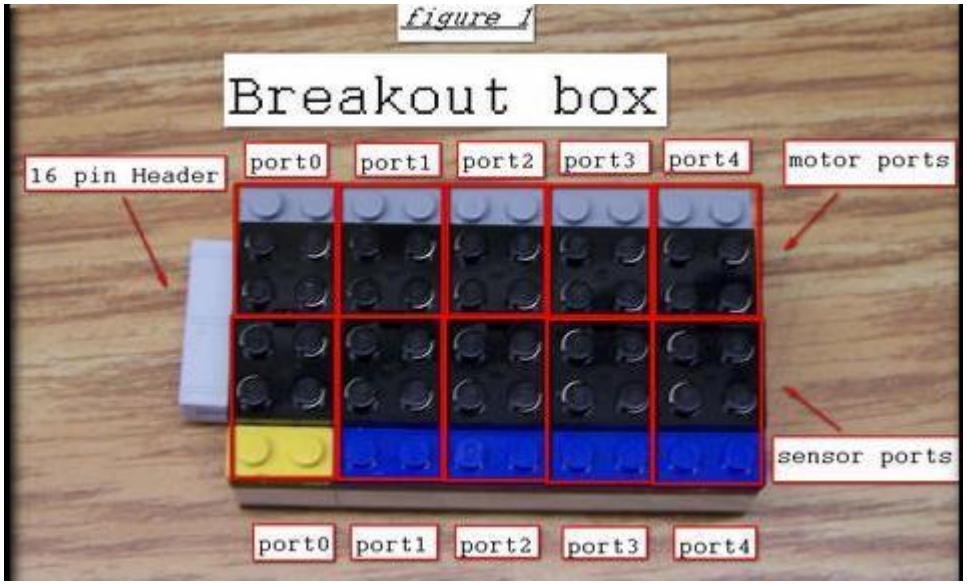
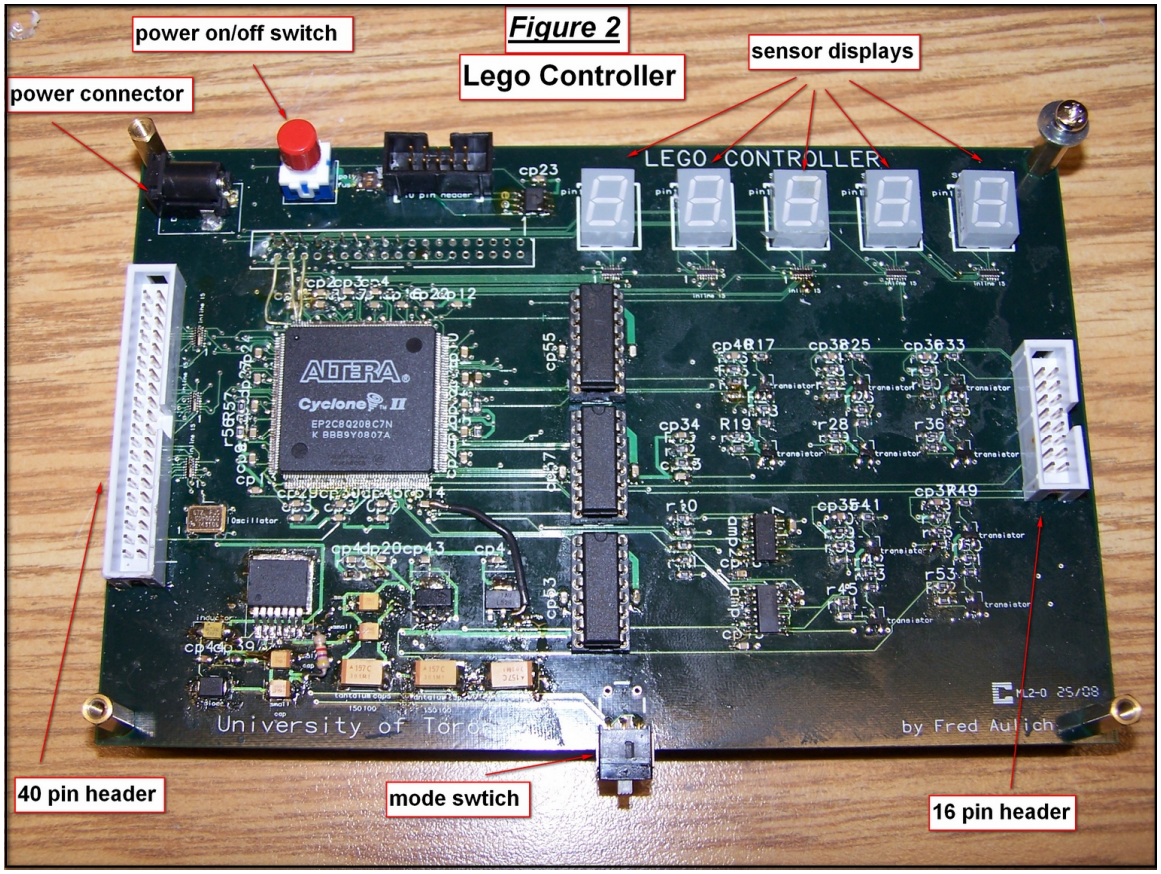
Lego Controller

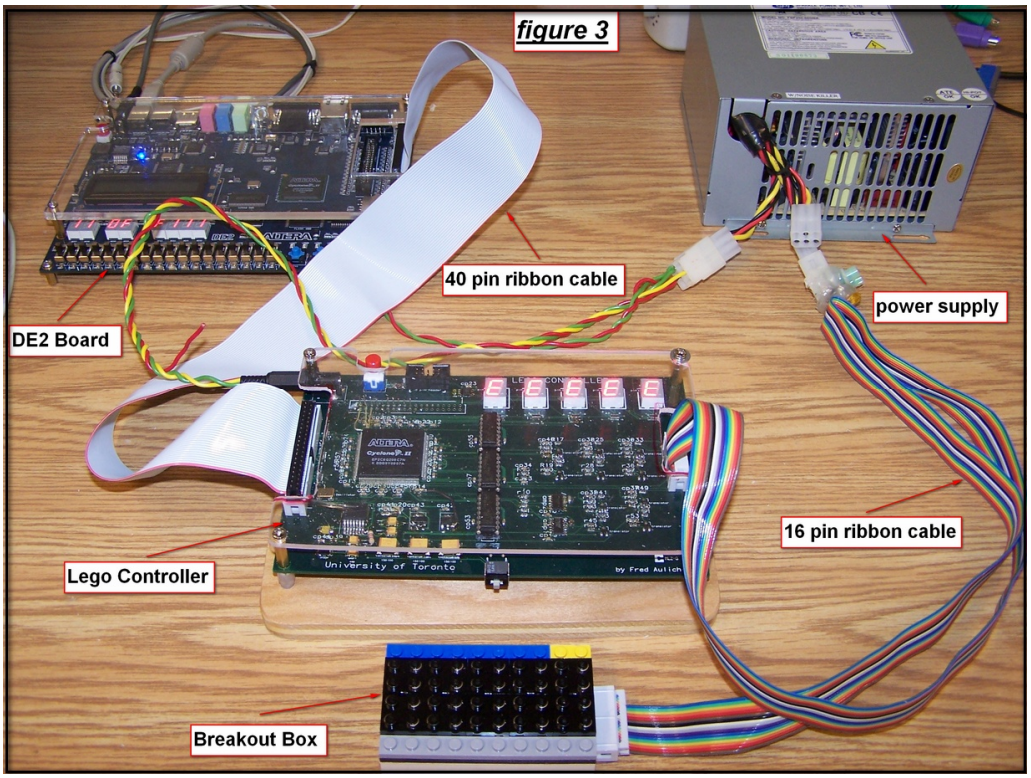
Version1.8

The University of Toronto Lego Controller consists of two parts, a **Breakout box** (*figure 1*) and a **Lego controller** (*figure2*)





The 16 pin header on the Lego controller connects the Breakout box to the Lego controller. The 40 pin header connects the Lego controller to the DE2 board. The full setup looks like (*figure 3*)



I) **Breakout box**

The Breakout box has 5 connections for sensors and 5 connections for the motors. Each 2X2 black Lego block represents one connection to either a sensor or a motor. The 2X2 black blocks along the blue 1x8 plate and yellow 1x2 plate are used to connect to the **sensors**. The black 2X2 block beside the short yellow plate represents sensor 0. The others follow in numerical order. The 2X2 black blocks that are along the gray 1x10 plate are for the **motors**. The 2X2 block directly opposite the yellow 1X2 plate is motor 0. The others follow in numerical order.

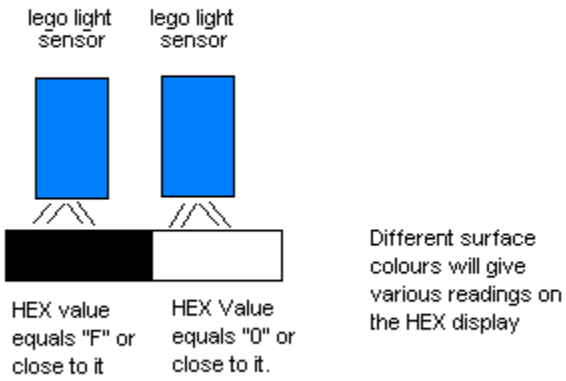
II**) Lego controller**

The Lego controller is an intelligent interface that communicates information between the NIOS processor on the DE2 board and the Lego motors and sensors. There are 5 HEX displays on the Lego controller. Each HEX display is used to represent a sensor value that is being read from the Breakout box. A HEX value of "F" represents **no light** and a HEX value of "0" represents **full light**. The left most HEX display represents sensor 0. The rest follow sequentially sensor 1 to 4.

There are two ways to read the sensor values:

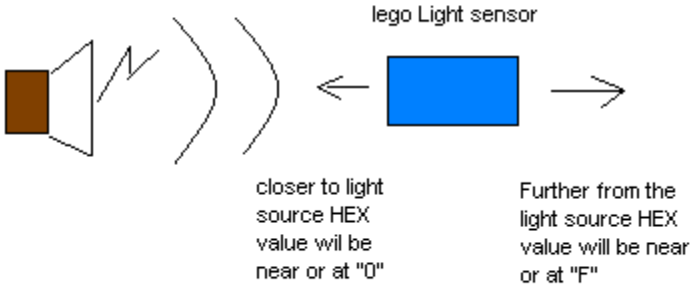
1) The sensor can be kept at a fixed distance and the surface colour/shade can be changed. A bright surface like yellow or white, will display a low HEX value. A darker surface colour like black or brown, will display a high HEX value. See (*figure 4*)

figure 4



2) Keep a fixed light source on the sensor and move the sensor to and from the light source. See (*figure 5*)

figure 5



III) **<u>40 pin header</u>**

The Lego controller communicates with the DE2 board via the **40 pin header**. The Lego controller uses 30 bits of the GPIO port on the NIOS processor. GPIO pins 0-9 are used to drive the motors. GPIO pins 10, 12,14,16,18 are used to enable reading and writing sensor values from the Lego controller. GPIO pins 11, 13,15,17,19 and 21-31 are used for

different control functions on the Lego controller. The bits marked **N/A** are not used. *Table 1* shows a breakdown of what each bit on the GPIO does.

Table 1

Motor Bits	**description**	**Bit = '1'**	**Bit = '0'**
GP0	Motor0	Disabled	Enabled
GP1	Motor0	Counter Clockwise	Clockwise
GP2	Motor1	Disabled	Enabled
GP3	Motor1	Counter Clockwise	Clockwise
GP4	Motor2	Disabled	Enabled
GP5	Motor2	Counter Clockwise	Clockwise
GP6	Motor3	Disabled	Enabled
GP7	Motor3	Counter Clockwise	Clockwise
GP8	Motor4	Disabled	Enabled
GP9	Motor 4	Counter Clockwise	Clockwise
Sensor bits	**description**	**Bit = '1'**	**Bit = '0'**
GP10	Sensor0	Disabled	Enabled
GP11	Ready Sensor0	Sensor data not valid	Sensor0 data valid
GP12	Sensor1	Disabled	Enabled
GP13	Ready Sensor1	Sensor data not valid	Sensor1 data valid
GP14	Sensor2	Disabled	Enabled
GP15	Ready Sensor2	Sensor data not valid	Sensor2 data valid
GP16	Sensor3	Disabled	Enabled
GP17	Ready Sensor3	Sensor data not valid	Sensor3 data valid
GP18	Sensor4	Disabled	Enabled
GP19	Ready Sensor4	Sensor data not valid	Sensor4 data valid
GP20	N/A	N/A	N/A
Control bits	**description**	**Bit = '1'**	**Bit = '0'**
GP21	Control	Polling	State
GP22	Control	Disabled	Load Threshold Data
GP23	Control	Threshold data	Threshold data
GP24	Control	Threshold data	Threshold data
GP25	Control	Threshold data	Threshold data
GP26	Control	Threshold data	Threshold data
GP27	Control	Polling /State **sensor 0**	Polling /State **sensor 0**
GP28	Control	Polling /State **sensor 1**	Polling /State **sensor 1**
GP29	Control	Polling /State **sensor 2**	Polling /State **sensor 2**
GP30	Control	Polling /State **sensor 3**	Polling /State **sensor 3**
GP31	Control	State **sensor 4**	State **sensor 4**

3.1) **Motor**

There are two bits to control the motor. One bit is for enabling the motor and the other is to determine the direction the motor turns. To drive a motor the enable bit must be '0' and depending on the value in the direction bit the motor will either go clockwise or counter clockwise.

3.2) **Sensors**

The sensor select bits GP10, 12,14,16,18 have a dual purpose.
In **polling mode** if GP21 is '1' and one of the select bits GP10, 12,14,16,18 is '0' then the 4 bit value stored at GP27-30 will be read from the Lego controller sensor data register. **Note** you can only read one sensor at a time.
In **Threshold mode** if GP22 is '0' and one of the select bits GP10,12,14,16,18 is '0' then the 4 bit value stored at GP23-26 will be read by the Lego controller sensor threshold register. **Note** you can only load one sensor at a time.

3.3) **Control**

GP21 is used to select between **polling** mode or **state** mode.

Polling mode (GP21 = '1')
It is *important* to note that in polling mode you can only read one 4 bit sensor value at a time from GP27-30. By setting one of the bits in GP10, 12,14,16,18 to '0', it will enable the NIOS processor to read the 4 bit sensor value at GP27-30.

GP10 is '0' then read **sensor 0** value at GP27-30
GP12 is '0' then read **sensor 1** value at GP27-30
GP14 is '0' then read **sensor 2** value at GP27-30
GP16 is '0' then read **sensor 3** value at GP27-30
GP18 is '0' then read **sensor 4** value at GP27-30

Important note*

When reading data you must check the corresponding **sensor data valid bit** for each sensor to ensure it is **low** (**Valid)**. If it is not low the data value stored at GP27-30 may not be valid.

Valid data bit for;
Sensor0 is **GP11**
Sensor1 is **GP13**
Sensor2 is **GP15**
Sensor3 is **GP17**
Sensor4 is **GP19**

State mode (GP21 ='0')
Depending on what the 4 bit threshold value is, the state value of GP27-31 is either '1' or '0'. For example, consider that the 4 bit threshold value of **sensor 0** has been preloaded to HEX 9,(Binary1001). If the sensor value at **sensor 0** is greater than or equal to HEX 9 then **GP27** is '1' otherwise it is '0'. It is *important* to note that for proper threshold readings, the 4 bit threshold value GP23-26 for all sensors must be preloaded.

GP22 is used to load the 4 bit threshold HEX value stored at GP23-26 into the Lego controller threshold register. If **GP22** is '0' and any of the sensor enable bits **GP10, 12,14,16,18** are '0'then the 4 bit threshold HEX value stored at **GP23-26** is loaded into the 4 bit threshold register on the Lego controller.

If bits **GP10** and **GP22** are '0', then threshold value stored at **GP23-26** will be loaded to **Sensor 0** Lego controller threshold register.
If bits **GP12** and **GP22** are '0', then threshold value stored at **GP23-26** will be loaded to **Sensor 1** Lego controller threshold register.
If bits **GP14** and **GP22** are '0', then threshold value stored at **GP23-26** will be loaded to **Sensor 2** Lego controller threshold register.
If bits **GP16** and **GP22** are '0', then threshold value stored at **GP23-26** will be loaded to **Sensor 3** Lego controller threshold register.
If bits **GP18** and **GP22** are '0', then threshold value stored at **GP23-26** will be loaded to **Sensor 4** Lego controller threshold register.

Note: Each threshold value must be loaded one at a time to the Lego controller. You do not need to read the corresponding **ACK** bit for each sensor in this mode

GP23-26 are used to store the 4 bit threshold value to be loaded into the Lego controller register. Whatever value is stored on these pins will be transferred to the Lego controller threshold register.

GP27-GP30 have a dual purpose:
GP31 is only used in state mode:

In **state** mode (GP21 = '0') if the preloaded value in the 4 bit threshold register is **greater than** the sensor value detected by the Lego controller then the sensor bit is '0' otherwise it is '1'.

Sensor 0 = GP27 State bit
Sensor 1 = GP28 State bit
Sensor 2 = GP29 State bit
Sensor 3 = GP30 State bit
Sensor 4 = GP31 State bit

In order to use interrupts with the Lego controller you must run the Lego controller in state mode. The bits GP27-31 on the NIOS processor must be configured as interrupt inputs. To do this read the section in the NIOS handout that deals with setting up interrupts. .

In **polling** mode these 4 bits represent the HEX value that is displayed on the HEX display depending on which sensor is selected GP10,12,14,16,18.

IV) **Slider Switch**

The Lego controller has a **3 position slider** switch located at the bottom centre of the Lego controller board.

When the switch is in the **centre** position "**NIOS Mode**" then the Lego controller will act as a passive device and communicate with the NIOS GPIO ports JP1/2.
********NOTE***** The switch must be in centre position in order for the Lego controller to properly communicate with the NIOS processor.***

If the switch is in the **right** position "**Motor Test Mode"** then the Lego controller will act as an active device and tests the Lego motors. It is a quick test to see if all the motors are working. *******NOTE**** this test will only work when the mode switch is in the right position and the 40 pin ribbon cable is disconnected from the DE2 board.*** All motors are tested. The test goes through a repeating sequence where each motor is driven clockwise then counter clockwise then all the motors go clockwise and finally all the motors go counter clockwise.

If the switch is in the **left** position "**Sensor Test Mode"** then the Lego controller will act as an active device and test the Lego sensors. *******NOTE**** this test will only work when the mode switch is in the left position and the 40 pin ribbon cable is disconnected from the DE2 board.*** In this mode if the sensor value goes below *HEX* value '9' then the motor will rotate. Sensor 1 turns motor1 on, sensor 2 turns motor 2 on and so on.