

# Race the PicoPi — Lesson 3

Learn how to move your PicoPi motors according to Infrared Reflection Sensors. Then you can make it follow a black line around a race track.

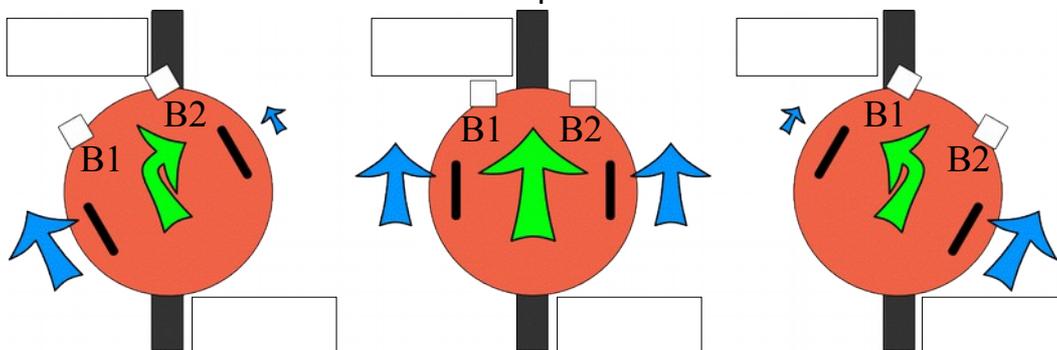
## How it Works:

Infra-red light is sent downwards by an IR-LED and some light is reflected by the track surface. The phototransistor collects this light and then the microchip's ADC (Analog to Digital Converter) gets a value between 0-255.

Note: Infra-red light is invisible to your eyes but is still 'seen' by electronic devices such as digital cameras and the photo-transistors on the PicoPi.

## Main Activity:

Control the PicoPi with the PicoFlow Alpha software to read two IR sensors

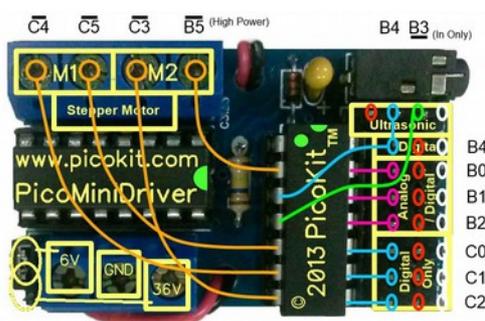


## TIPS:

- Plug the Reflection sensors into the Analog pins (B1 & B2) each with a 3-pin cable. Warning: the B1 sensor interferes with the PicoFlow USB Programmer, so disconnect it each time before you program the PicoPi.
- Use the tools from last lesson but replace the File Tool with an Analog Input tool. Then the motor will update its speed according to the sensor.
- Add a 2nd Motor Tool and Analog Input Tool to make the PicoPi follow a line.

## HINT:

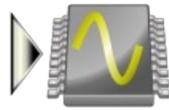
- Use the diagram above to help match each Analog Input Tool (B1 or B2) for each motor direction at the correct speed. 'Invert' changes the speed.



**PicoMiniDriver** — Control Board



**PicoPi—Robot** (designed by David Oswald & Philip Tallents)



# Analog Input Tool

## About the Analog Input Tool:

The Analog Input Tool can measure the voltage of analog pins on certain Micro-controllers and places the result on a selected File Register.

## Selecting a PIC for your project: (1)



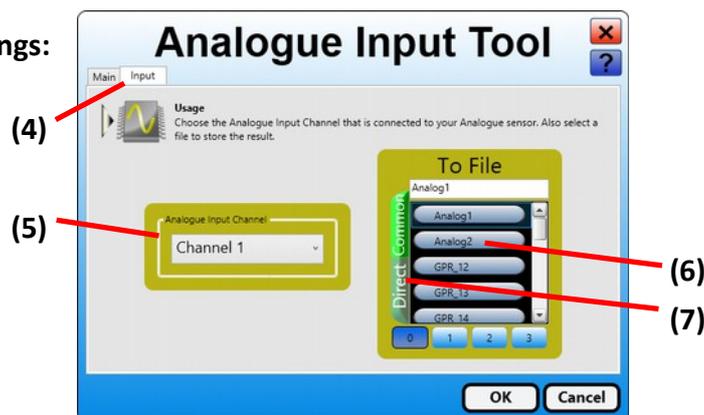
The PicoPi uses a PIC called the PIC16F506<sup>(1)</sup>. This device can be selected in PicoFlow Alpha on the Programming Panel of the Ribbon Bar.

## Linking the Analog Input Tool: (2)



The Analog Input Tool must be linked up in a loop<sup>(2)</sup> so the value updates regularly. **Warning:** Incorrect Linking with the Analog Tool out of the loop<sup>(3)</sup> will not vary the speed of the motor.

## Analog Input Tool Settings:



The diagram above shows the **Input** tab<sup>(4)</sup> of the Analog Input Tool. Select the analog channel/pin<sup>(5)</sup> to measure from the Analog Input Channel. These pins are the Analog pins B0, B1, & B2 on the PicoMiniDriver.

**Warning:** B0 & B1 conflict with the USB Programmer so always disconnect B0 & B1 analog sensors before programming and then reconnect them afterwards. **Note:** B2 is not affected.

On the **To File** panel, choose the **Direct** memory button<sup>(7)</sup>, then any available blue General Purpose Register<sup>(12)</sup> **GPR\_12** and rename it in the textbox<sup>(6)</sup> to something memorable eg. *Analog1*; **Note:** spaces are not allowed here.

Click **OK**<sup>(14)</sup> to save your changes and exit back to the Designer window.

- TIPS:**
- Tick the Annotate Enable checkbox<sup>(9)</sup> and then write a comment in the Annotation textbox<sup>(10)</sup> to say which Analog Sensor is used.
  - Plug in the PicoFlow USB Programmer to the PicoPi and Program<sup>(1)</sup> it.





# Output Tool

## About the Output Tool:

The Output Tool can control the states of all the pins on the selected PIC Micro-controller.

## Selecting a PIC for your project: (1)



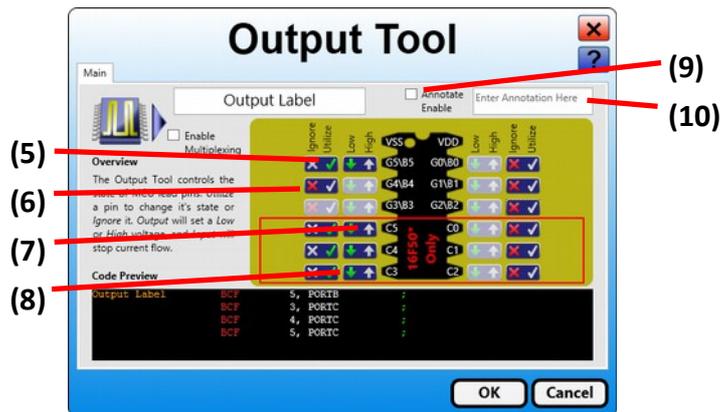
The PicoPi uses a PIC called the PIC16F506<sup>(1)</sup>. This device can be selected in PicoFlow Alpha on

## Linking the Output Tool:



The Output Tool must be linked up<sup>(2)</sup> from the Start Tool and at the end the links must loop<sup>(3)</sup> back to the Output Tool. If you don't link it up at the end, then you will receive an error<sup>(4)</sup>.

## Output Tool Settings:



The diagram above shows the black PIC microchip with all its pins. These pins are labelled with a letter and a number, such as B5. Each pin has two buttons to control what it does.

Button 1  <sup>(5)</sup> controls if the pin can also be Utilized (tick), or Ignored (cross).

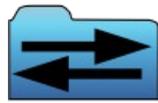
While button 2 <sup>(8)</sup> sets the pin of the PIC with a *High* (5.5V) or *Low* (0V) voltage :

- Utilize <sup>(5)</sup>
- Ignore <sup>(6)</sup>
- High <sup>(7)</sup>
- Low <sup>(8)</sup>

Both High and Low are considered outputs because they pass electrical *current* through the pin. High *sources* (puts) current onto the pin and Low *sinks* (takes) current away from the pin.

- TIPS:**
- Tick  the pins you want to use, and Cross  the pins you don't.
  - Tick the Annotate Enable checkbox<sup>(9)</sup> and then write a comment in the Annotation textbox<sup>(10)</sup> to say the direction the PicoPi moves in.





# File Tool

## About the File Tool:

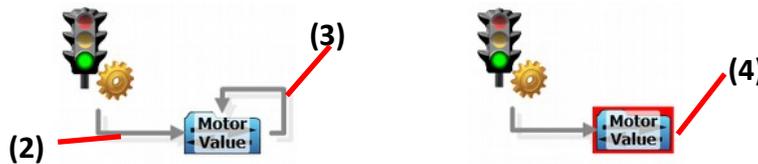
The File Tool transfers number values into memory Registers of your selected PIC Micro-controller. Different PIC Micro-controllers have different amounts of memory Registers.

## Selecting a PIC for your project: (1)



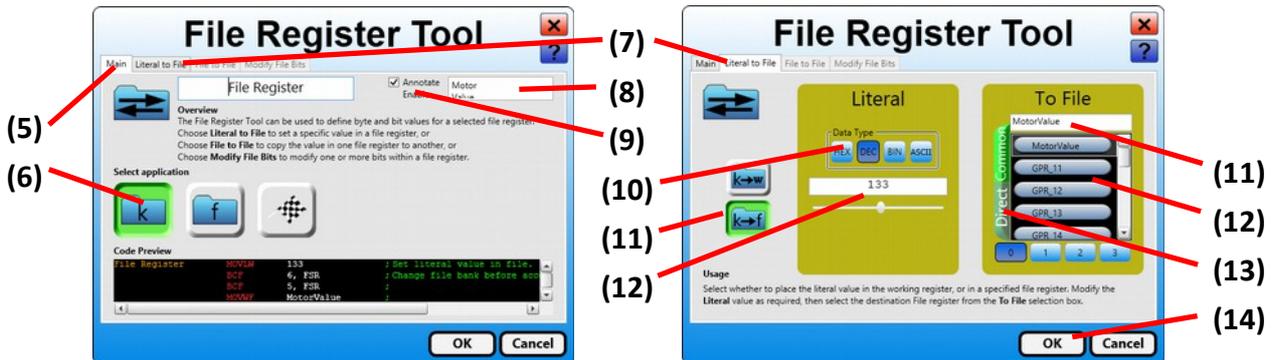
The PicoPi uses a PIC called the PIC16F506<sup>(1)</sup>. This device can be selected in PicoFlow Alpha on the Programming Panel of the Ribbon Bar.

## Linking the File Tool:



Generally, the File Tool is used after the Start Tool or whenever a value must be changed. Link the File Tool up from the Start Tool<sup>(2)</sup> and at the end the links must loop<sup>(3)</sup> back to the File Tool. If you don't link it up at the end, then you will receive an error<sup>(4)</sup> when programming.

## File Tool Settings:



The diagrams above show two tabs of the File Register Tool. On the Main tab<sup>(5)</sup> select the  button<sup>(6)</sup> then choose the **Literal to File** tab<sup>(7)</sup>.

On the Literal to File tab<sup>(7)</sup> the  button<sup>(11)</sup> lets you to transfer a number constant to a file register.

Choose the **DEC** button<sup>(10)</sup> on the **Literal** panel and then enter a value (0-255) in the textbox<sup>(12)</sup>. On the **To File** panel, choose the **Direct** memory button<sup>(13)</sup>, then any available blue General Purpose Register<sup>(12)</sup> eg.  and rename it in the textbox<sup>(11)</sup> to something memorable eg. *MotorValue*; **Note:** spaces are not allowed here.

Click **OK**<sup>(14)</sup> to save your changes and exit back to the Designer window.

The File Tool doesn't so much by itself because you can't see it working until it is connected to the Motor Tool on the next page.

- TIPS:**
- Go back into the File Tool Settings and update the Annotation. Tick the **Annotate Enable** checkbox<sup>(9)</sup> and then write a comment in the Annotation textbox<sup>(8)</sup> to indicate the affect of the tool. In this case you could use the name and new value of the File Register. eg. *MotorVal<sup>(enter)</sup>=133*



# Motor Tool

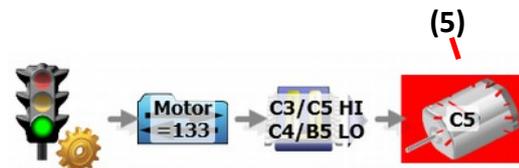
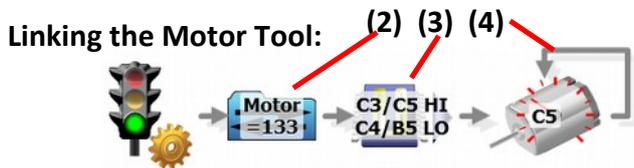
## About the Motor Tool:

The Motor Tool can control the average voltage at one pin on the PIC Micro-controller using PWM (Pulse Width Modulation). Examples 25%  50%  75% 

## Selecting a PIC for your project: (1)

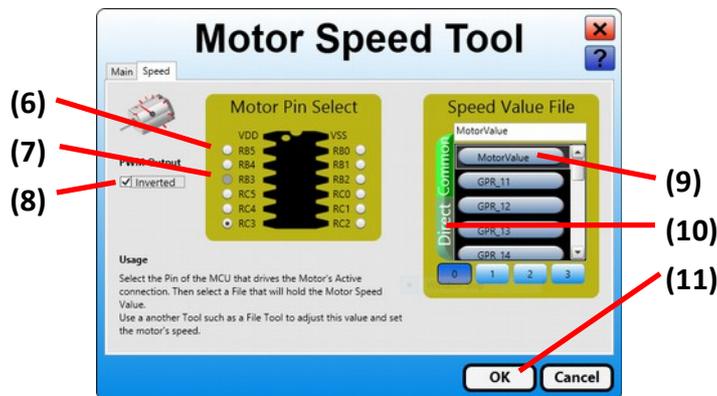


The PicoPi uses a PIC called the PIC16F506<sup>(1)</sup>. This device can be selected in PicoFlow Alpha on the Programming Panel of the Ribbon Bar.



Generally, the Motor Tool must be used with an Output Tool<sup>(3)</sup> and File Tool<sup>(2)</sup> all linked up from the Start Tool and at the end the links must loop<sup>(4)</sup> back to the Motor Tool. If you don't link it up at the end, then you will receive an error<sup>(5)</sup>.

## Motor Tool Settings:



The diagram above shows the black PIC microchip with all its pins. These pins are labelled with a letter and a number, such as B5. Any output pin<sup>(6)</sup> can be selected (input pins<sup>(7)</sup> are greyed out). You will also need to choose the same Speed Value File that you used in the File Tool by selecting the **Direct** memory button<sup>(10)</sup>, then the blue **MotorValue** GPR<sup>(9)</sup>.

Click **OK**<sup>(11)</sup> to save your changes and exit back to the Designer window.

Both High and Low are considered outputs because they pass electrical *current* through the pin. High *sources* (puts) current onto the pin and Low *sinks* (takes) current away from the pin.

## TIPS:

- Check a pin that controls the L293D Motor Driver (B5, C3, C5, or C4). One of these pins will control either motor (M1 or M2) and its direction.
- If a large number in the File Tool makes the Motor drive slowly, then you can check the Inverted Checkbox<sup>(8)</sup> to invert the speed.
- Plug in the PicoFlow USB Programmer to the PicoPi and Program<sup>(1)</sup> it. 

# 3. Race the PicoPi—Worksheet

**Revision Questions:** What does ADC stand for and explain in your own words how it works?

Explain how the reflection sensor works?

How does the Motor Tool get the *Speed Value* and how does it get updated?

How does the Analog Input Tool affect each of the motors when turning?

In line following, explain what happens to a motor when a sensor goes over the

**Explore Further:** Use the view finder of a digital camera to observe the IR LEDs on the reflection sensors. Turn the PicoPi on, and comment on the appearance of these LEDs.

Research the Wavelength and Frequency of Infrared Light and comment on why it can't be seen by human eyes.

Use a Multimeter to measure the voltage on the yellow wire of each Reflection sensor cable. Record the results for both sensors over black and white surfaces.