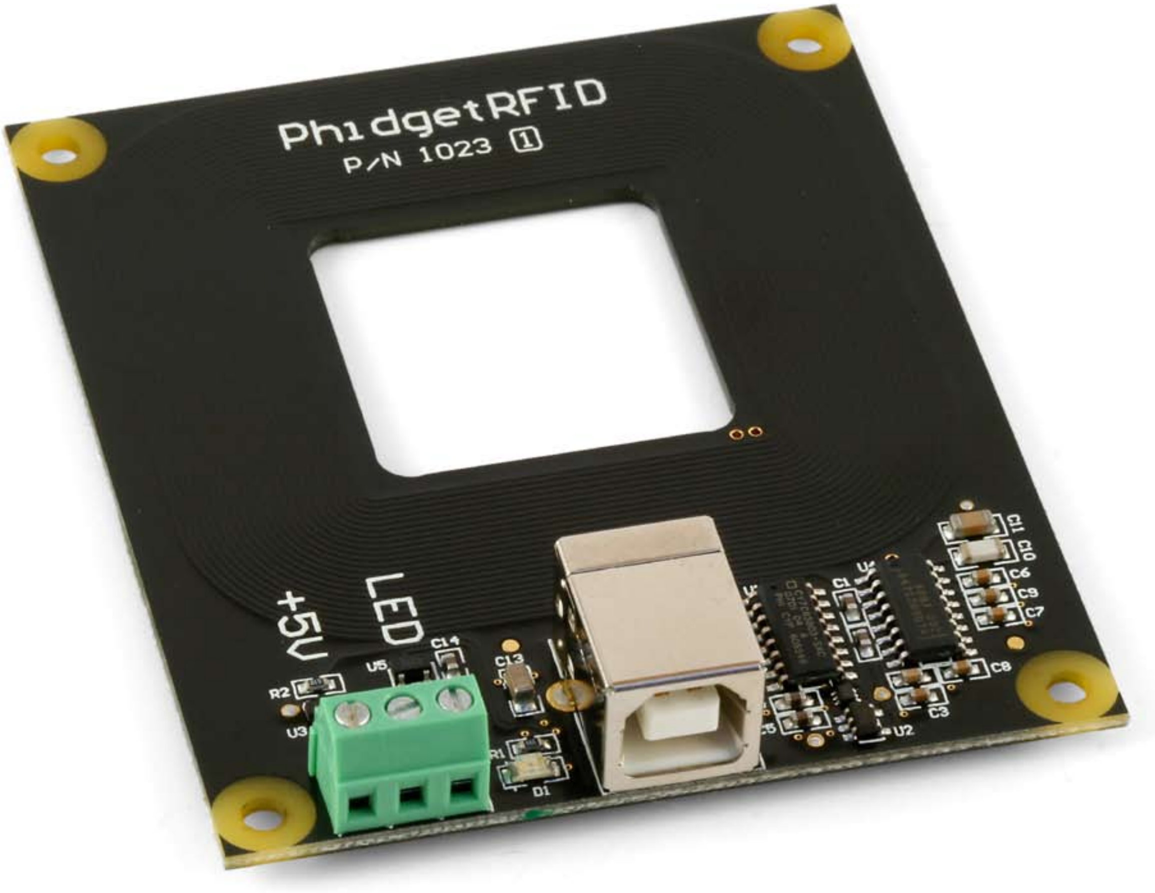
PhidgetRFID



Operating Systems:

Windows 2000/XP/Vista, Windows CE, Linux, and Mac OS X

Application Programming Interfaces (APIs):

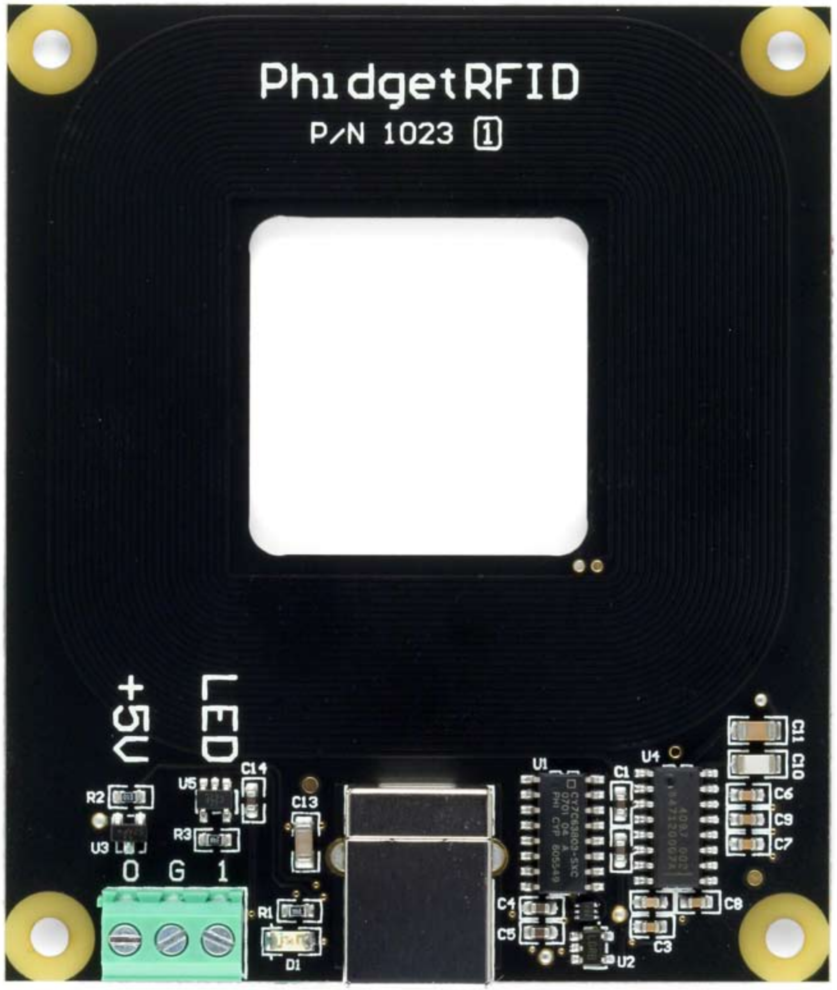
Visual Basic, VB.NET, C, C++, C#, Flash 9, Flex, Java, LabVIEW, and Matlab

Examples:

You will find program examples in the download section of www.phidgets.com

What Can the PhidgetRFID Do?

The PhidgetRFID allows you to read unique identification strings stored within RFID tags. It provides a generic, convenient way to use your PC for Radio Frequency IDentification.



RFID Tags

The PhidgetRFID can be used with any RFID tag designed for the EM4102 protocol. RFID tags come in a variety of shapes and sizes to suit various applications. All RFID tags sold by Phidgets are guaranteed to be unique, and are available as:

- 30mm Disc Tags (these can be sewn into garments, attached to objects)
- Credit Card Sized Tags (good for security identification applications)
- Key Fob Tags (attach easily to key rings)

Visit www.phidgets.com for more information.

Getting Started

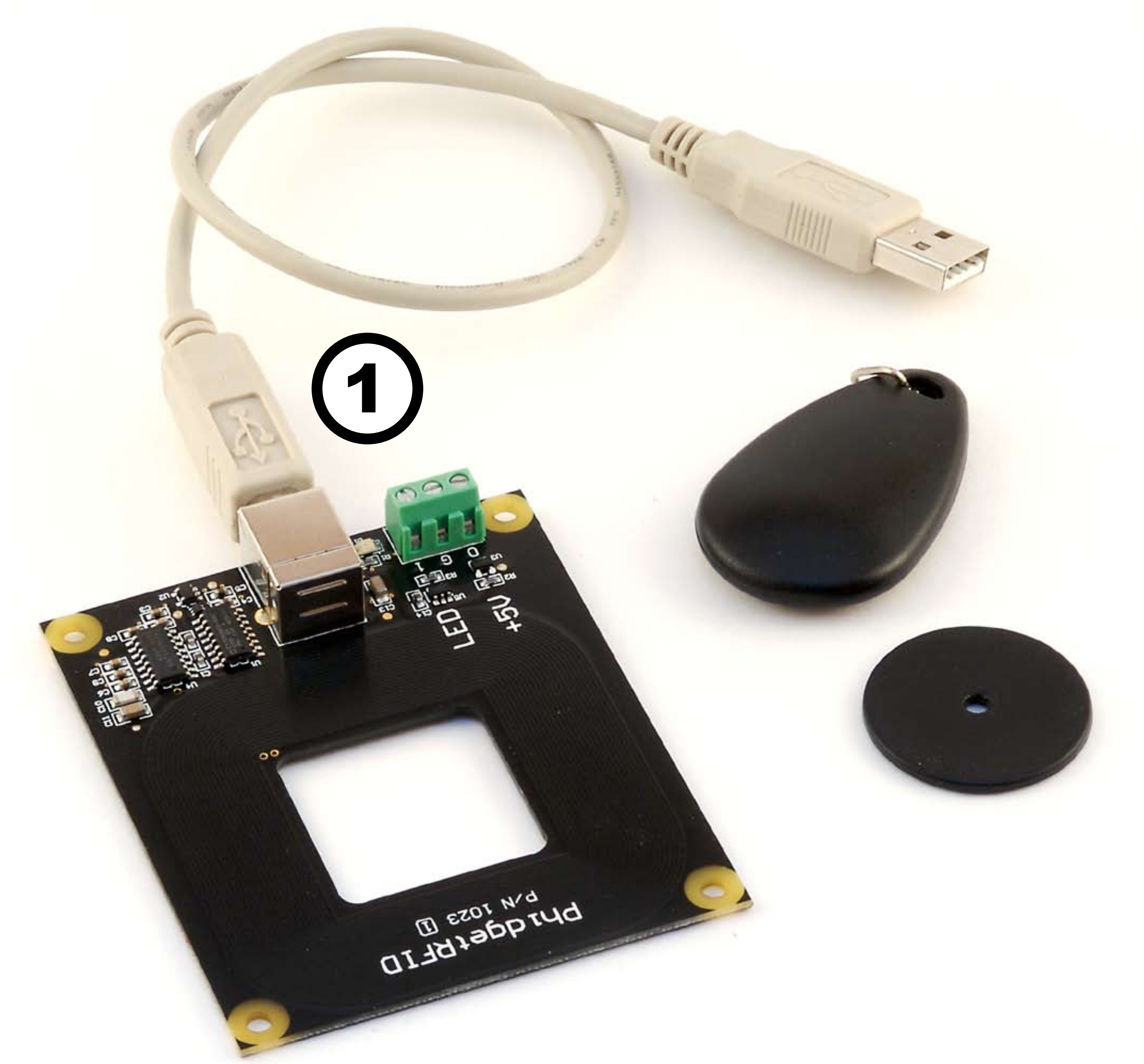
Installing the hardware

The kit contains:

- A PhidgetRFID
- A USB Cable

You will also need:

- A compatible RFID tag



1. Connect the PhidgetRFID board to the computer using the USB cable.

Download and Install the software

Go to www.phidgets.com >> downloads

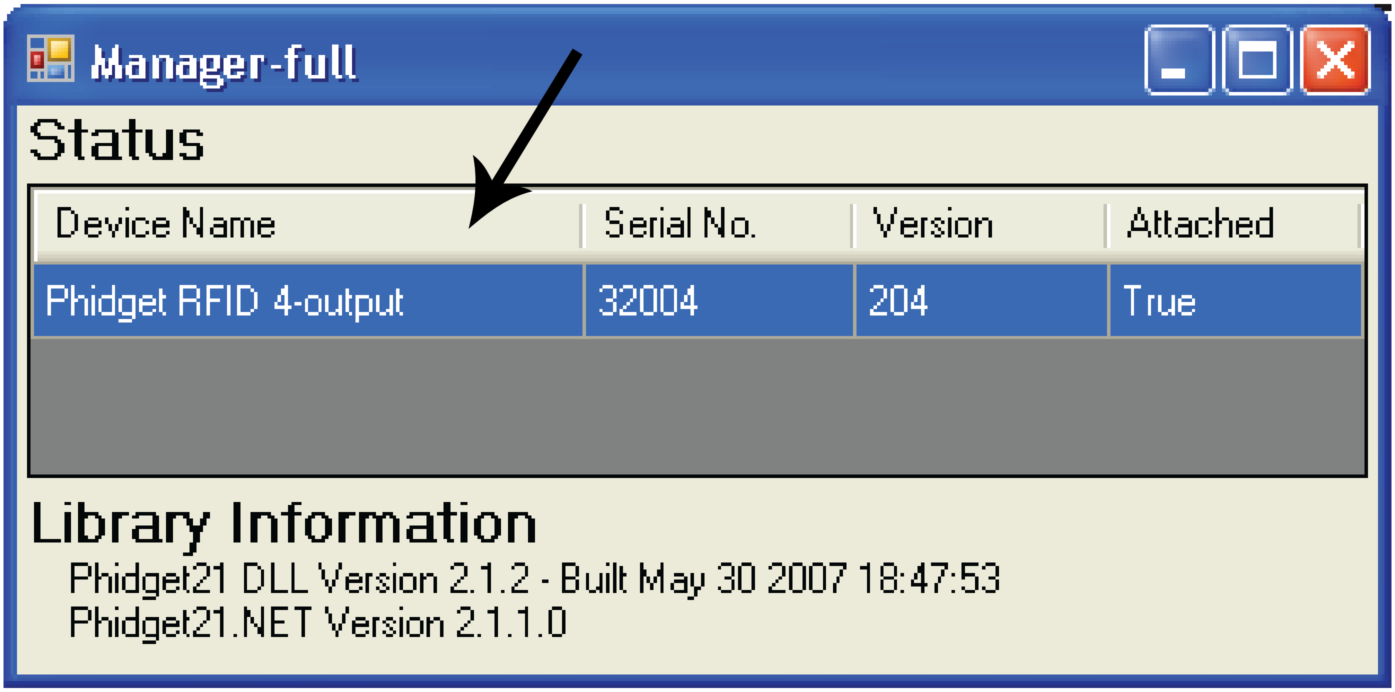
Select your operating system (Windows, Linux, MAC OS)

Select the language you want to use and download the appropriate examples and Libraries.

Install the Libraries and decompress the Example file.

Testing the PhidgetRFID using Windows

- Note that some examples are not available for Linux, Mac OSX or Windows CE.
- Make sure that you have installed the libraries and decompressed your example file.



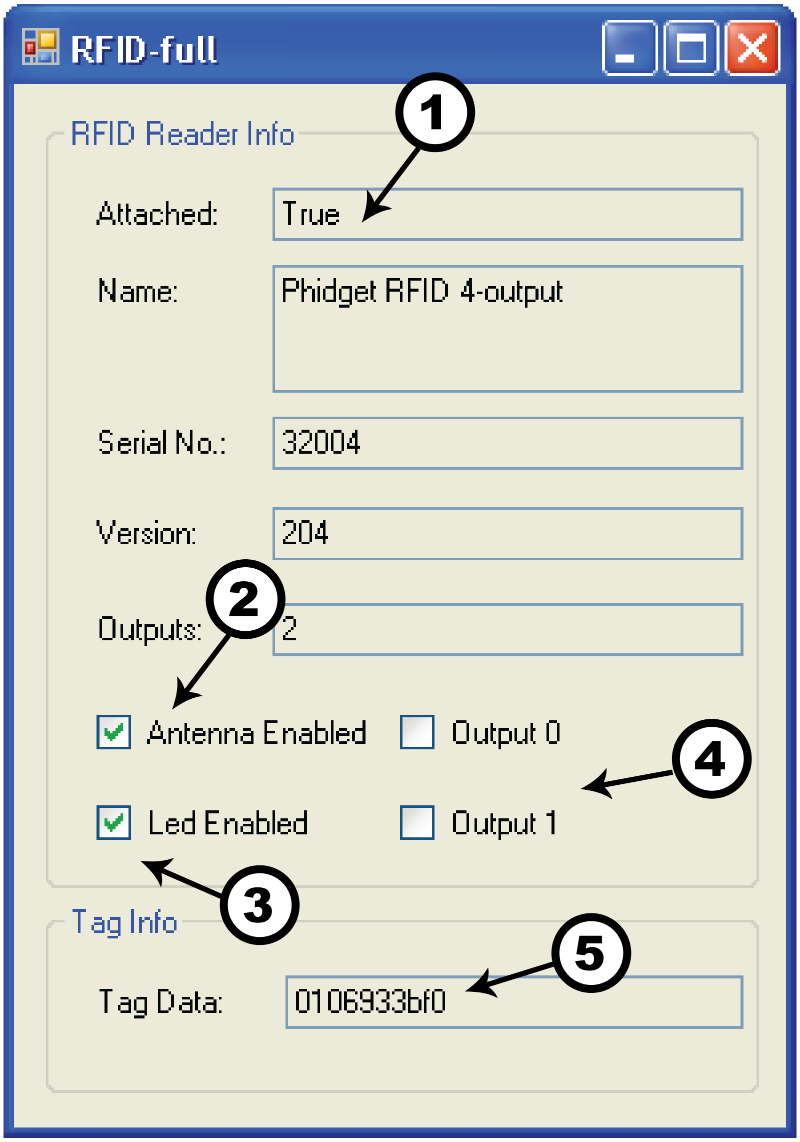
Status

Device Name	Serial No.	Version	Attached
Phidget RFID 4-output	32004	204	True

Library Information

Phidget21 DLL Version 2.1.2 - Built May 30 2007 18:47:53
Phidget21.NET Version 2.1.1.0

Run the program ***Manager-full*** to make sure that the ***PhidgetRFID*** is properly connected to your PC.



RFID Reader Info

Attached: True

Name: Phidget RFID 4-output

Serial No.: 32004

Version: 204

Outputs: 2

☑ Antenna Enabled □ Output 0

☑ Led Enabled □ Output 1

Tag Info

Tag Data: 0106933bf0

1. Run the program ***RFID-full*** and check that the box labelled Attached contains the word True.
2. Click on the Antenna Enabled box to enable the antenna.
3. Click on the LED Enabled box to turn the LED on.
4. The Output 0 box controls the +5V digital output and the Output 1 box controls the external LED digital output.
5. Bring an RFID tag close to the PhidgetRFID board and check that the identification string is displayed. Make sure that the Antenna is enabled for this step.

Programming a Phidget

Where to get information

- Go to www.phidgets.com >> downloads
- Select the Operating System and the language you want to use.
- Download the appropriate API manual and read the section under the ***RFID*** heading.
- Have a look at the source code of the ***RFID-full*** program.
- Have a look at the C# example below.
- Modify an existing program or write your own program from scratch.

Simple example written in C#

```
/* - RFID simple -
 * This program simply displays the data that is generated by an RFID phidget in a very simple case and outputs
 * it to the console.  This simple example covers the basics of connecting and using an RFID phidget.  For a
 * more detailed example, see RFID-full.
 *
 * Please note that this example was designed to work with only one Phidget RFID connected.
 * For an example using multiple Phidget RFIDs, please see a “multiple” example in the RFID Examples folder.
 *
 * Copyright 2007 Phidgets Inc.
 * This work is licensed under the Creative Commons Attribution 2.5 Canada License.
 * To view a copy of this license, visit http://creativecommons.org/licenses/by/2.5/ca/
 */
using System;
using System.Collections.Generic;
using System.Text;
using Phidgets; //Needed for the RFID class and the PhidgetException class
using Phidgets.Events; //Needed for the phidget event handling classes

namespace RFID_simple
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                RFID rfid = new RFID(); //Declare an RFID object

                //initialize our Phidgets RFID reader and hook the event handlers
                rfid.Attach += new AttachEventHandler(rfid_Attach);
                rfid.Detach += new DetachEventHandler(rfid_Detach);
                rfid.Error += new ErrorEventHandler(rfid_Error);

                rfid.Tag += new TagEventHandler(rfid_Tag);
                rfid.TagLost += new TagEventHandler(rfid_TagLost);
                rfid.open();

                //Wait for a Phidget RFID to be attached before doing anything with the object
                Console.WriteLine(“waiting for attachment...”);
                rfid.waitForAttachment();

                //turn on the antenna and the led to show everything is working
                rfid.Antenna = true;
                rfid.LED = true;

                //keep waiting and outputting events until keyboard input is entered
                Console.WriteLine(“Press any key to end...”);
                Console.Read();

                //turn off the led
                rfid.LED = false;

                //close the phidget and dispose of the object
                rfid.close();
                rfid = null;
                Console.WriteLine(“ok”);
            }
            catch (PhidgetException ex)
            {
                Console.WriteLine(ex.Description);
            }
        }

        //attach event handler...display the serial number of the attached RFID phidget
        static void rfid_Attach(object sender, AttachEventArgs e)
        {
            Console.WriteLine(“RFID reader {0} attached!”, e.Device.SerialNumber.ToString());
        }

        //detach event handler...display the serial number of the detached RFID phidget
        static void rfid_Detach(object sender, DetachEventArgs e)
        {
            Console.WriteLine(“RFID reader {0} detached!”, e.Device.SerialNumber.ToString());
        }

        //Error event handler...display the error description string
        static void rfid_Error(object sender, ErrorEventArgs e)
        {
            Console.WriteLine(e.Description);
        }

        //Print the tag code of the scanned tag
        static void rfid_Tag(object sender, TagEventArgs e)
        {
            Console.WriteLine(“Tag {0} scanned”, e.Tag);
        }

        //print the tag code for the tag that was just lost
        static void rfid_TagLost(object sender, TagEventArgs e)
        {
            Console.WriteLine(“Tag {0} lost”, e.Tag);
        }
    }
}
```

Learning more ...

- check out the forums

- check out the Phidgets projects

Technical Section

RFID

RFID (radio frequency identification) systems use data strings stored inside RFID tags (or transponders) to uniquely identify people or objects when they are scanned by an RFID reader. These types of systems are found in many applications such as passport protection, animal identification, inventory control systems, and secure access control systems.

RFID Protocols

In order for an RFID reader like the PhidgetRFID to communicate with an RFID tag, they must share a common protocol. This protocol acts as a set of rules for the way data is transmitted wirelessly between the reader and tag. The PhidgetRFID (as well as RFID tags sold by Phidgets) uses the EM4102 protocol. Any other tags that also use the EM4102 protocol can be used with the PhidgetRFID.

Communication and Effectiveness

RFID tags come in two main varieties: passive and active. Active tags have their own power supply which they use to power an antenna to communicate and transmit data. Passive tags derive the power they require to operate directly from the RF output of the RFID reader, and no other power supply is necessary. This makes passive tags cheaper to produce and easier to implement.

Because passive tags require a strong RF field to operate, their effective range is limited to an area in close proximity to the RFID reader. In the case of the PhidgetRFID, tags brought within approximately 3" of the reader can be read. The distance over which the RFID tag is usable is affected by such things as the tag shape and size, materials being used in the area near the reader, and the orientation of the reader and tag in respect to each other and in their operating environment. The smaller a tag, the closer it must be to the reader to operate.

Multiple Readers

Multiple PhidgerRFIDs within 1 to 2 meters may interfere with eachother. This can be overcome in software by enabling the antennae of individual PhidgetRFID readers in sequence.

1. Starting with all readers disabled, enable the antenna of the first PhidgetRFID reader.
2. Wait for 100ms or more to detect any tags.
3. Disable the antenna of the first reader and enable the antenna of the second, and perform another wait cycle.

Multiple Tags

The PhidgetRFID offers no capability for collision detection or collision avoidance. If two tags are brought within the read field of a PhidgetRFID reader at the same time, neither tag will be read. An RFID tag should be removed from the read field before a second tag is introduced.

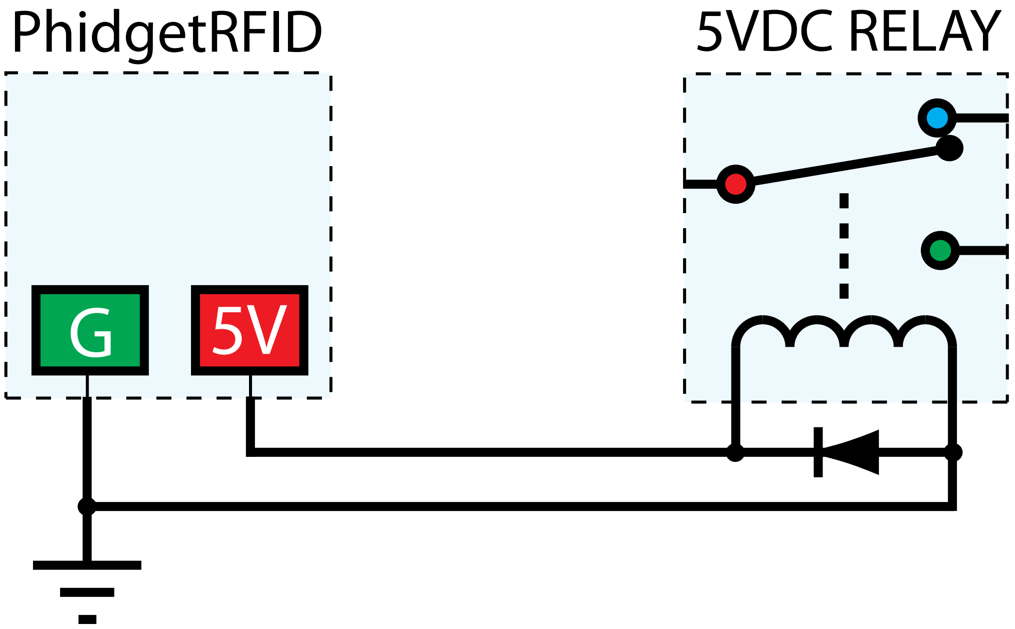
Controlled Outputs

The PhidgetRFID has four outputs - two of which are available to the user, and two of which are for internal control of the Phidget board only.

Output 0 is a +5V source from the USB bus through a P-Channel MOSFET with less than one ohm impedance. This can be used to switch a TTL or CMOS device, or it can be used to drive a 5VDC relay such as the Aromat JS1-5V. Output 1 is an LED drive output at 5VDC with approximately 20mA of available current (250 ohm CMOS output). Both Output 0 and 1 are available in hardware at the terminal blocks on the PhidgetRFID board. If Output 0 is used to drive a relay, a fast clamping diode must be placed across the relay drive pins as shown in the diagram below. Not doing so can result in permanent damage to the PhidgetRFID board.

The PhidgetRFID comes equipped with an on-board LED that can be controlled using the LED property in software. Additionally, the on-board antenna can be enabled or disabled in software by using the RF Enable property. The antenna must be enabled for the PhidgetRFID to detect and read an RFID tag.

Output	Function	Connection
0	+5VDC Source	Terminal Block
1	External LED Drive	Terminal Block
LED	Internal LED Drive	Internal Only
RF Enable	RF Antenna Enable	Internal Only

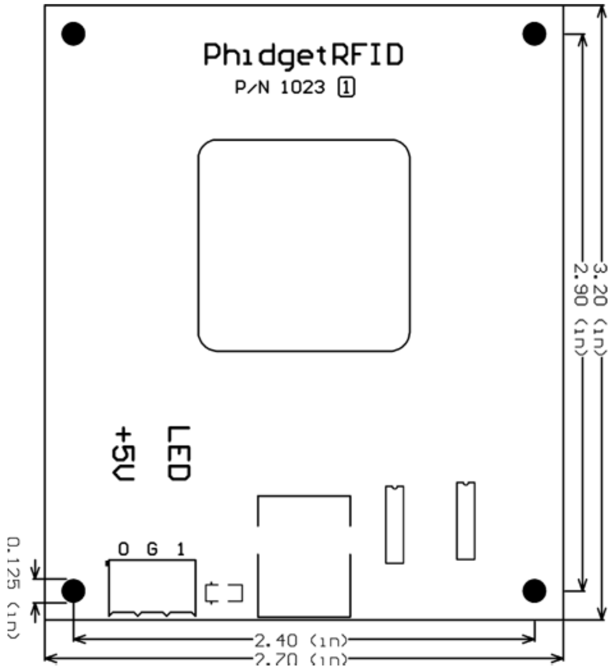


Device Specifications

Antenna Output Power (max, far field)	< 10 μW
Antenna Resonant Frequency	125kHz - 140kHz
Effective Read Range (typ)	3”
Communication Protocol	EM4102
Read Update Rate	30 updates / second
External +5V Supply Voltage	5VDC
External +5V Supply Current Limit	400mA
External LED Supply Voltage	5VDC
External LED Supply Current Limit	16mA
External LED Output Resistance	250 Ohms
Recommended Terminal Wire Size	16 - 26 AWG
Terminal Wire Strip Length	5 - 6mm (0.196” - 0.236”)
USB-Power Current Specification	500mA max
Device Quiescent Current Consumption	16mA
Device Active Current Consumption	100mA max

Mechanical Drawing

1:1 scale



Product History

Date	Product Revision	Comment
June 2002	DeviceVersion 100	Product Release.
April 2004	DeviceVersion 200	Onboard LED, 2 Digital Outputs added. Ability to enable/ disable Antenna added.
Jan 2005	DeviceVersion 201	RF Circuitry upgraded to improve reliability.
Jan 2006	DeviceVersion 202	Onboard microprocessor upgraded to Flash version.
Jun 2006	DeviceVersion 204	Low Voltage Reset set at 4.7 Volts
April 2007	PCB Rev 0, DeviceVersion 205	Protocol parsing bug fixed which garbled top 3 bits of RFID Tag.
July 2007	PCB Rev 1, DeviceVersion 206	Unused internal I/O pulled high out of an abundance of caution, bus current characterized. Terminal block moved to edge of PCB, center of antenna routed out, digital output transients on startup eliminated.