
arduino-rttl-player Documentation

Release 0.0.0

ponty

November 02, 2012

CONTENTS

1	About	2
2	Basic usage	3
3	Manual Installation	4
4	Automatic Installation	5
4.1	General	5
4.2	Ubuntu	5
4.3	Ubuntu uninstall	5
5	Examples	6
6	Simulation	8
7	Library size	9
8	Build tests	10
8.1	Results	10
9	Doxygen documentation	11

arduino-rttl-player

Date November 02, 2012

PDF [arduino-rttl-player.pdf](#)

ABOUT

RTTTL player library for [Arduino](#).

Links:

- home: <https://github.com/ponty/arduino-rtttl-player>
- documentation: <http://ponty.github.com/arduino-rtttl-player>

Features:

- based on RTTTL example in [Tone](#) library
- blocking mode only
- song can be either in PROGMEM or RAM
- support for both internal and external improved [Tone](#) library
- build tests
- examples
- library size calculation
- simulation
- API documentation with doxygen
- header only library
- supported Arduino versions: 0022, 1.0

BASIC USAGE

```
//#include <Tone.h>
// if Tone.h is included before this include,
// then the external Tone library is used
// else the core tone()/noTone() functions.
#include <rtttl.h>

const int pinSpeaker = 13;
const int octave = 0;
const char song_P[] PROGMEM = 'Indiana:d=4,o=5,b=4000:e,8p,8f,8g,8p,1c6';

Rtttl player;

void setup(void)
{
    player.begin(pinSpeaker);
    player.play_P(song_P, octave);
}

void loop(void)
{
}
```

MANUAL INSTALLATION

<http://arduino.cc/en/Guide/Environment#libraries>

AUTOMATIC INSTALLATION

4.1 General

- install `arduino`
- install `confduino`
- install the library:

```
# as root
python -m confduino.libinstall https://github.com/ponty/arduino-rtttl-player/zipball/master
```

4.2 Ubuntu

```
sudo apt-get install arduino
sudo apt-get install python-pip
sudo pip install confduino
sudo python -m confduino.libinstall https://github.com/ponty/arduino-rtttl-player/zipball/master
```

4.3 Ubuntu uninstall

```
sudo python -m confduino.libremove rtttl
```


EXAMPLES

`./rtttl/examples/Progmem/Progmem.pde`

```
//#include <Tone.h>  // the core tone()/noTone() are used.  
#include <rtttl.h>
```

```
const int pinSpeaker = 13;  
const int octave = 0;
```

```
// this solution is recommended:  
// the song is stored in program memory only
```

```
const char song_P[] PROGMEM =  
    "Indiana:d=4,o=5,b=250:e,8p,8f,8g,8p,1c6,8p.,d,8p,8e,1f,p.,g,8p,8a,8b,8p,1f6,p,a,"
```

```
Rtttl player;
```

```
void setup(void)  
{  
    player.begin(pinSpeaker);  
  
    // player.play(song, octave);  
    player.play_P(song_P, octave);  
}
```

```
void loop(void)  
{  
}
```

`./rtttl/examples/ExtTone/ExtTone.pde`

```
#include <Tone.h> //the external Tone library is used  
#include <rtttl.h>
```

```
const int pinSpeaker = 13;  
const int octave = 0;
```

```
const char song_P[] PROGMEM =  
    "Indiana:d=4,o=5,b=250:e,8p,8f,8g,8p,1c6,8p.,d,8p,8e,1f,p.,g,8p,8a,8b,8p,1f6,p,a,"
```

```
Rtttl player;
```

```
void setup(void)  
{  
    player.begin(pinSpeaker);  
    player.play_P(song_P, octave);  
}
```

```
void loop(void)  
{  
}
```

./rtttl/examples/Ram/Ram.pde

```
//#include <Tone.h>  // the core tone()/noTone() are used.
#include <rtttl.h>
```

```
const int pinSpeaker = 13;
const int octave = 0;
```

```
// this solution is not recommended:
// the song is stored in program memory and then copied into RAM
```

```
const char song[] =
    "Indiana:d=4,o=5,b=250:e,8p,8f,8g,8p,1c6,8p.,d,8p,8e,1f,p.,g,8p,8a,8b,8p,1f6,p,a,
```

```
Rtttl player;
```

```
void setup(void)
{
    player.begin(pinSpeaker);

    player.play(song, octave);
}
```

```
void loop(void)
{
}
```

SIMULATION

Simavr is used for simulation

Code:

```
#include <rtttl.h>

const int pinSpeaker = 13;
const int octave = 0;

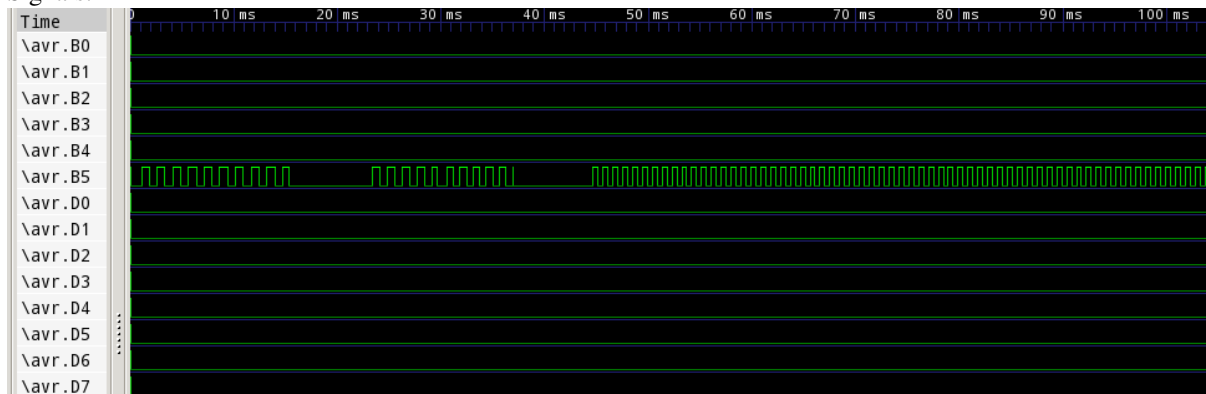
const char song_P[] PROGMEM = "Indiana:d=4,o=5,b=4000:e,8p,8f,8g,8p,1c6";

Rtttl player;

void setup(void)
{
    player.begin(pinSpeaker);
    player.play_P(song_P, octave);
}

void loop(void)
{
}
```

Signals:



LIBRARY SIZE

Comment	Code snippet	Program bytes	Data bytes
no song	<code>player.play(0);</code>	1248	7
no song	<code>player.play_P(0);</code>	1248	7
song in RAM	<code>player.play("Indiana:d=4,o=5,b=4000:e,8p,8f,8g,8p,1c6");</code>	1290	49
song in PROGMEM	<code>player.play_P(PSTR("Indiana:d=4,o=5,b=4000:e,8p,8f,8g,8p,1c6"));</code>	1288	7

The maximum size is calculated as a difference:

Program1 = empty template + code snippet

Program2 = empty template

Maximum library size = Program1 size - Program2 size

Actual size can be lower. MCU=atmega168

Template:

```
#include <rtttl.h>

Rtttl player;

const int pinSpeaker = 13;

void setup()
{
    Serial.begin(9600);
    tone(5, 400); // to include tone lib

    snippet;
}

void loop()
{
}
```

BUILD TESTS

8.1 Results

8.1.1 Arduino version 0022

index	board	Progmem	ExtTone	Ram
1	atmega8	OK (P:3446 D:32)	OK (P:3658 D:35)	OK (P:3446 D:226)
2	atmega48	OK (P:3738 D:39)	BIG (P:4148 D:42)	OK (P:3738 D:233)
3	atmega168	OK (P:3858 D:39)	OK (P:4266 D:42)	OK (P:3858 D:233)
4	atmega328p	OK (P:3858 D:39)	OK (P:4266 D:42)	OK (P:3858 D:233)
5	atmega640	OK (P:4688 D:60)	OK (P:4608 D:42)	OK (P:4688 D:254)
6	atmega1280	OK (P:4904 D:60)	OK (P:5892 D:63)	OK (P:4904 D:254)
7	atmega2560	OK (P:4908 D:60)	OK (P:4840 D:42)	OK (P:4908 D:254)

8.1.2 Arduino version 1.0

index	board	Progmem	ExtTone	Ram
8	atmega8	OK (P:3454 D:32)	ERR	OK (P:3454 D:226)
9	atmega48	OK (P:3746 D:39)	ERR	OK (P:3746 D:233)
10	atmega168	OK (P:3868 D:39)	ERR	OK (P:3868 D:233)
11	atmega328p	OK (P:3868 D:39)	ERR	OK (P:3868 D:233)
12	atmega640	OK (P:4700 D:60)	ERR	OK (P:4700 D:254)
13	atmega1280	OK (P:4718 D:60)	ERR	OK (P:4718 D:254)
14	atmega2560	OK (P:4722 D:60)	ERR	OK (P:4722 D:254)

DOXYGEN DOCUMENTATION

Files