arduino-rttl-player Documentation

Release 0.0.0

ponty

February 10, 2013

CONTENTS

arduino-rttl-player

Date February 10, 2013

PDF arduino-rttl-player.pdf

CHAPTER ONE

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
- hardware: check 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

CHAPTER

TWO

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)
{
}
```

CHAPTER

THREE

MANUAL INSTALLATION

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

CHAPTER

FOUR

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
```

CHAPTER

FIVE

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,8

Rtttl player;

void setup(void)
{
        player.begin(pinSpeaker);
        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,8

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)
{
}
```

CHAPTER

SIX

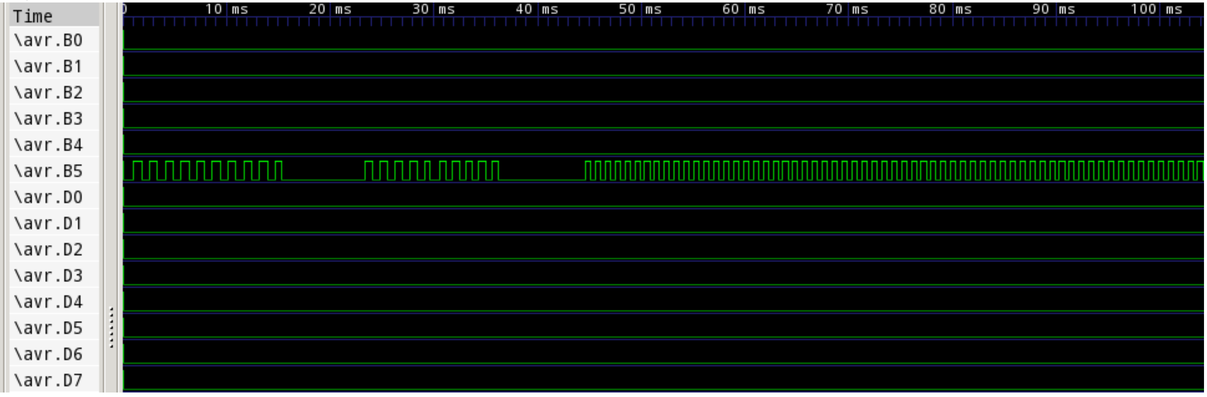
SIMULATION

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:



CHAPTER

SEVEN

LIBRARY SIZE

Comment	Code snippet	Program bytes	Data bytes
no song	`player.play(0);`	986	1
no song	`player.play_P(0);`	962	1
song in RAM	`player.play("Indiana:d=4,o=5,b=4000:e,8p,8f,8g,8p,1c6");`	1026	41
song in PROGMEM	`player.play_P(PSTR("Indiana:d=4,o=5,b=4000:e,8p,8f,8g,8p,1c6"));`	1002	1

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()
{
}
```

CHAPTER

EIGHT

BUILD TESTS

8.1 Results

8.1.1 Arduino version 0022

MCU	Progmem	ExtTone	Ram
atmega8	OK (P:3442 D:32)	OK (P:3636 D:35)	OK (P:3442 D:226)
atmega48	OK (P:3794 D:39)	BIG (P:4186 D:42)	OK (P:3794 D:233)
atmega168	OK (P:3856 D:39)	OK (P:4246 D:42)	OK (P:3856 D:233)
atmega328p	OK (P:3856 D:39)	OK (P:4246 D:42)	OK (P:3856 D:233)
atmega640	OK (P:4938 D:60)	OK (P:4840 D:42)	OK (P:4938 D:254)
atmega1280	OK (P:4902 D:60)	OK (P:5872 D:63)	OK (P:4902 D:254)
atmega2560	OK (P:4906 D:60)	OK (P:4820 D:42)	OK (P:4906 D:254)

8.1.2 Arduino version 1.0

CHAPTER

NINE

DOXYGEN DOCUMENTATION

Files