How the lora.begin(name) function works

July 10th 2015

Contents

1 Usage

To change the name of an arduino object, you must use the

```
void LoraShield::begin(String dns);
```

function after calling the init() function. For example:

```cpp
#include <LoraShield.h>
#include <SPI.h>

LoraShield lora;
void setup()
{
  lora.init();
  String name = "beta.s.ackl.io";
  lora.begin(name);
}

void loop()
{ }
```

2 Arduino side

This function is defined in *LoraShield.cpp*:

```
/**
 * \description: send the name of the object to the shield
 * \param: name - the name
 */
void LoraShield::begin(String name)
{
  digitalWrite(SS_PIN, LOW);

  // The byte is the status of the last command
  int previous_cmd_status = SPI.transfer(ARDUINO_CMD_HOSTNAME);
  delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);

  // Data size to be sent
  // MSB
  int shield_status = SPI.transfer(name.length() >> 8);
  delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);

  shield_status = SPI.transfer(dns.length());
  delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);

  //Send:  payload as bytes to send
  for (int i = 0; i < name.length() ; i++)
  {
    shield_status = SPI.transfer(name[i]);
    delayMicroseconds(WAIT_TIME_BETWEEN_BYTES_SPI);
  }

  digitalWrite(SS_PIN,HIGH);
  delay(WAIT_TIME_BETWEEN_SPI_MSG*2);
}
```

So the contiki board will receive a packet like :

0x21	length_msb	length_lsb	name

3   Contiki side

3.1   Change the name

The parsing of the previous packet is defined in the
/platform/lorafabian/apps/arduino_ interface/arduino_ cmd.c file. The program waits a AR-
DUINO_CMD_HOSTNAME and write the new name in the file: /HOSTNAME_ LORA.

3.2   Get the current name

The current name is written in the file /HOSTNAME_ LORA. This is an example of a function
which get the current name (updateHOSTNAME() in the file
/examples/lorafabian/lorafab_ beacon_ answer_ new/lorafab_ beacon_ answer_ new.c):

```
#include "cfs/cfs.h"

char coap_payload_beacon[150];
```

```c
/**
 * \brief: update the hostname with /HOSTNAME file
 */
void updateHOSTNAME()
{
  char dns[150];//The content of the file
  int fd;
  //Read in /HOSTNAME_LORA
  fd = cfs_open("/HOSTNAME_LORA", CFS_READ);
  if(fd >= 0) {
    //Read 500 char
    cfs_read(fd, dns, sizeof(dns));
    cfs_close(fd);
    //Get the real hostname
    int size = 0;
    //Because the space significate the end of the hostname
    while(dns[size] != '\0')
      ++size;
    //final = the real url
    char final[size];
    int i;
    for(i = 0; i != sizeof(final) +1; ++i)
      final[i] = dns[i];
    strcpy(coap_payload_beacon, "{\"n\":\"");
    strcat(coap_payload_beacon, final);
    strcat(coap_payload_beacon, "\"}");
  }
  else {
    printf("READING ERROR\n\r");
    strcpy(coap_payload_beacon, "{\"n\":\"default.test\"}");
    return;
  }
  printf("HOSTNAME : %s\n\r", coap_payload_beacon);
}
```

Note: if the *HOSTNAME_ LORA* doesn't exists, the default name will be *default.test*.