/\*

\* Getting Started example sketch for nRF24L01+ radios

\* This is an example of how to send data from one node to another using data structures

\* Updated: Dec 2014 by TMRh20

\*/

//lightstart

int lightPin = A1; // select the input pin for the potentiometer

//int ledPin = 13; // select the pin for the LED

// Min 0

// Max 1024

//lightend

//water begin

#define WATER\_SENSOR 7

#include <SPI.h>

#include "RF24.h"

byte addresses[][6] = {"1Node","2Node"};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* User Config \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\* Set this radio as radio number 0 or 1 \*\*\*/

bool radioNumber = 1;

/\* Hardware configuration: Set up nRF24L01 radio on SPI bus plus pins 7 & 8 \*/

RF24 radio(9,10);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Used to control whether this node is sending or receiving

bool role = 1;

/\*\*

\* Create a data structure for transmitting and receiving data

\* This allows many variables to be easily sent and received in a single transmission

\* See http://www.cplusplus.com/doc/tutorial/structures/

\*/

struct dataStruct{

unsigned long \_micros;

int lightSensorValueA;

bool waterA;

int lightSensorValue;

bool water;

}myData;

void setup() {

Serial.begin(115200);

Serial.println(F("RF24/examples/GettingStarted\_HandlingData"));

Serial.println(F("\*\*\* PRESS 'T' to begin transmitting to the other node"));

pinMode(WATER\_SENSOR, INPUT);

radio.begin();

// Set the PA Level low to prevent power supply related issues since this is a

// getting\_started sketch, and the likelihood of close proximity of the devices. RF24\_PA\_MAX is default.

radio.setPALevel(RF24\_PA\_LOW);

// Open a writing and reading pipe on each radio, with opposite addresses

if(radioNumber){

radio.openWritingPipe(addresses[1]);

radio.openReadingPipe(1,addresses[0]);

}else{

radio.openWritingPipe(addresses[0]);

radio.openReadingPipe(1,addresses[1]);

}

//myData.value = 0;

Serial.println("Value set to 0");

// Start the radio listening for data

radio.startListening();

}

void loop() {

//water start

if(digitalRead(WATER\_SENSOR) == HIGH)

myData.water = true;

else

myData.water = false;

//water end

myData.lightSensorValueA = analogRead(lightPin);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Ping Out Role \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if (role == 1) {

radio.stopListening(); // First, stop listening so we can talk.

Serial.println(F("Now sending"));

// Serial.print("myData.sensorValue: ");

// Serial.println(myData.sensorValue);

myData.\_micros = micros();

if (!radio.write( &myData, sizeof(myData) )){

Serial.println(F("failed due to write"));

}

// Serial.print(F("Size of myData: "));

// Serial.println(sizeof(myData));

radio.startListening(); // Now, continue listening

unsigned long started\_waiting\_at = micros(); // Set up a timeout period, get the current microseconds

boolean timeout = false; // Set up a variable to indicate if a response was received or not

while ( ! radio.available() ){ // While nothing is received

if (micros() - started\_waiting\_at > 200000 ){ // If waited longer than 200ms, indicate timeout and exit while loop

timeout = true;

break;

}

}

if ( timeout ){ // Describe the results

Serial.println(F("Failed, response timed out."));

}else{

// Grab the response, compare, and send to debugging spew

radio.read( &myData, sizeof(myData) );

unsigned long time = micros();

// Spew it

// Serial.print(F("Sent at: "));

// Serial.print(time);

// Serial.print(F(", Got response myData.\_micros: "));

// Serial.print(myData.\_micros);

// Serial.print(F("\n Round-trip delay time-myData.\_micros: "));

// Serial.print(time-myData.\_micros);

Serial.print(F("lightsensorValueA: "));

Serial.println(myData.lightSensorValueA);

Serial.print(F("Water SensorA: "));

Serial.println(myData.waterA);

//Serial.print(F(", Size of myData: "));

//Serial.println(sizeof(myData));

}

// Try again 1s later

delay(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Pong Back Role \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if ( role == 0 )

{

if( radio.available()){

// Variable for the received timestamp

while (radio.available()) { // While there is data ready

radio.read( &myData, sizeof(myData) ); // Get the payload

}

Serial.println("Stop Listening");

radio.stopListening(); // First, stop listening so we can talk

// Serial.print("myData.sensorValue: ");

// Serial.println(myData.sensorValue);

// myData.value += 1; // Increment the float value

radio.write( &myData, sizeof(myData) ); // Send the final one back.

radio.startListening(); // Now, resume listening so we catch the next packets.

// Serial.print(F("Sent response \_micros: "));

//Serial.print(myData.\_micros);

Serial.print(F(" : lightsensorValue: "));

Serial.println(myData.lightSensorValue);

Serial.print(F(" : Water Sensor: "));

Serial.println(myData.water);

Serial.print(F(" : lightsensorValueA: "));

Serial.println(myData.lightSensorValueA);

Serial.print(F(" : Water SensorA: "));

Serial.println(myData.waterA);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Change Roles via Serial Commands \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if ( Serial.available() )

{

char c = toupper(Serial.read());

if ( c == 'T' && role == 0 ){

Serial.print(F("\*\*\* CHANGING TO TRANSMIT ROLE -- PRESS 'R' TO SWITCH BACK"));

role = 1; // Become the primary transmitter (ping out)

}else

if ( c == 'R' && role == 1 ){

Serial.println(F("\*\*\* CHANGING TO RECEIVE ROLE -- PRESS 'T' TO SWITCH BACK"));

role = 0; // Become the primary receiver (pong back)

radio.startListening();

}

}

} // Loop