octopus synchronous

# What

We were working together with the master students Nick & David and they had a problem with sending the data from anchor nodes to the root node. Basically they visualize the nodes in the network by taking the RSSI from the mobile node in reference to the anchor nodes. The anchor nodes send their data with the AMSend interface to the mobile node, which receives these packets with the AMReceive interface. The localization is controlled by the root node, so the mobile node forwards the received packets to the root. We investigated the problem and came to the conclusion that the data from the anchor nodes wasn’t being sent by the mobile node to the root node. The local data of the mobile node was being forwarded.

The solution is to synchronize the application with the help of some bools. So, we are going to implement a system that switches between the transmission of the data from the mobile node (local data) and the transmission of the data from the anchor nodes (external data)

# Application octopus

## bools

|  |
| --- |
| bool SWITCHTOB = FALSE;  bool BSEND = FALSE; |

We specify 2 bools to switch between the transmission of local & external data:

* SWITCHTOB is a bool which gives the control to the event Blindreceive so that it is able to transmit the data from the anchor nodes
* BSEND is a bool that we used in Timer.fired to make the sending process synchronous with the timer

## Timer.fired

|  |
| --- |
| if(voltageIsRead && lightIsRead && tempIsRead && humidityIsRead && rssiIsRead)  {  localCollectedMsg.count++;  if(root)  post serialSendTask();  else if(anchor)  post anchorSendTask();  else  if (SWITCHTOB == FALSE)  post collectSendTask();  else  BSEND = TRUE;  }    }  } |

Timer.fired is the central event in the application. We have added some lines of code to make the program synchronous. If we have collected the data from the sensors, and if the bool SWITCHTOB = FALSE then we transmit the local data. If it is TRUE then we switch to the blindreceive event and transmit the external data.

## collection send

|  |
| --- |
| event void CollectSend.sendDone(message\_t\* msg, error\_t error) {  if (error != SUCCESS)  reportProblem();  sendBusy = FALSE;    if (BSEND == FALSE)  {  SWITCHTOB = TRUE;  printf("Collectsend.sendDone van BLIND is gedaan\n");  call PrintfFlush.flush();  }  else  {  BSEND = FALSE;  SWITCHTOB = FALSE;  printf("Collectsend.sendDone van BROADCAST is gedaan\n");  call PrintfFlush.flush();  }  localCollectedMsg.reply = NO\_REPLY;  reportSent();  } |

We transmit the data with the CollectSend.send as a result we get the event CollectSend.sendDone. In this event we adjust the value of the bools to switch between the transmission of the local and external data. So, if we just transmitted the local data, then the bool BSEND == FALSE and thus we make the bool SWITCH equal to TRUE, so that when the timer fires, the bool BSEND become TRUE and we can transmit the external data.