Designing a Double LoRa Connectivity for the Arduino Portenta H7

Daniel López Pino*, Felix Freitag*, Mennan Selimi†,
* Department of Computer Architecture. Technical University of Catalunya. Barcelona, Spain
daniel.lopez.pino@upc.edu, felix.freitag@upc.edu
† Max van der Stoel Institute, South East European University. North Macedonia
m.selimi@seeu.edu.mk

***Abstract*—Machine learning is moving to the smallest computing devices. Today machine learning is applied even in tiny IoT microcontroller boards. In the IoT, LoRa is a popular communication technology to connect remote devices with gateways. Still, the confluence of machine learning in microcontrollers and networked LoRa connectivity is not yet fully exploited. In this paper we design a new LoRa connectivity for the Arduino Portenta H7, a recent microcontroller board equipped with embedded sensors suitable for diverse machine learning tasks. With the solution that we found the Arduino Portenta H7 is able to become part of a LoRa mesh network. This capacity increases the Portenta's range of applications. For the vision of distributed machine learning at the tiny edge, we can add with the Portenta an important board to become a smart compute node within a LoRa mesh network.**

***Index Terms*—IoT, machine learning, LoRa**

I. Introduction

Machine learning (ML) is nowadays deployed in ever smaller computing devices including microcontrollers found in Internet of Things (IoT) nodes [1]. Signal processing and ML tasks are performed directly at the IoT device, communicating only the result of a classification to remote gateways instead of the raw data [2].

LoRa is a popular communication technology for the IoT. LoRa links can cover several km of distance between the end node, which produces the data, and the gateway. Most IoT applications that use LoRa apply the LoRaWAN architecture [3]. This architecture builds a star topology which establishes a single hop between the end nodes and the gateway. LoRaWAN, however, does not directly interconnect the end nodes between themselves.

LoRa mesh networks have been proposed for diverse scenarios which cannot be addressed well by the LoRaWAN architecture [4]. For geographically spread IoT nodes and low gateway density, multi-hop LoRa can be used where intermediate nodes operate as repeaters that broadcast traffic to other LoRa nodes to finally reach a gateway [5]. Another scenario are gateway-less applications such as Meshtastic [6]. In this real application the nodes communicate only within the LoRa mesh network, without any gateway to the Internet.

Given the potential of LoRa mesh networks, in this paper we investigate the possibility of the Arduino Portenta H7 for becoming a node of a LoRa mesh network. The Portenta H7 is a recent microcontroller board which targets high performance industrial machine learning applications, being equipped for this with a dual core and very complete on-board sensors. We focus on the Portenta's LoRa connectivity challenge given that the board in practical applications has already been shown to be well prepared for running embedded machine learning applications[1].

II. Design

A. Analysis of the Arduino LoRa Vision Shield

The Arduino Portenta H7 is a board with two cores, a Cortex M7 running at 480 MHz and a Cortex M4 running at 240 MHz, to which shields can be connected in order to extend the board with additional features such as sensors. The shield which provides LoRa connectivity is the Arduino Portenta LoRa Vision Shield[2]. This shield contains the Murata CMWX1ZZABZ-078 module[3], which is internally composed of the SX1276 LoRa radio chip from Semtech, together with other elements for signal modulation, reception and transmission and the STM32L0 microcontroller.

Our aim is to analyze if the Arduino Portenta LoRa Vision Shield supports RadioLib[4], a popular communication library which allows to implement LoRa point-to-point communications. From analyzing the Murata datasheet we observed, however, that even though the module has SPI connectivity, it is not physically implemented in the Arduino Portenta Vision Shield and therefore the Arduino Portenta H7 cannot communicate directly with the Murata's SX1276 LoRa radio through this communication protocol. This fact disables the use of RadioLib which does require the SPI communication.

While we found that the use of RadioLib was not possible, the MKRWAN library is officially compatible with the Portenta LoRa Vision Shield. Using the MKRWAN library we observed that this library communicates with the Murata module over AT commands, which allows to manage the LoRaWAN connectivity of the radio. For operating AT commands, the serial bus named UART8 in the Portenta Vision Shield is connected

[1] https://docs.edgeimpulse.com/docs/development-boards/arduino-portenta-h7

[2] https://store.arduino.cc/products/arduino-portenta-vision-shield-lora

[3] https://www.murata.com/products/connectivitymodule/lpwa/overview/lineup/type-abz-078

[4] https://github.com/jgromes/RadioLib

 DOI 10.1109/LANMAN54755.2022.9820092

to the pins named Serial3 of the Portenta board. Therefore, the firmware implemented in the Murata module requires the radio to be connected to a LoRaWAN network. However, with AT commands as the only access to the module, it is not possible to conduct operations below the LoRaWAN layer, which is needed when we want to use the radio in a LoRa mesh network. In addition, it was pointed out that the current implementation of the MKRWAN library does not support some relevant AT commands[5].

B. *The choice of the double LoRa connectivity*

Our analysis reveled that while the Murata firmware supports LoRaWAN, it does not allow to operate on the layer below to manage basic LoRa packets. There have been efforts by the community to replace the Murata firmware, but the result is experimental and there is no official support for the Portenta LoRa Vision Shield[6],

We therefore chose the option to connect a second LoRa radio to the Arduino Portenta H7 and connect it as shown in Figure 1. The SCK, MISO, MOSI and CS signal connections belong to the SPI protocol. In addition, the connection with the SX1276 requires the interrupt and reset signals to correctly handle sending and receiving messages (INT and RST signals, respectively).

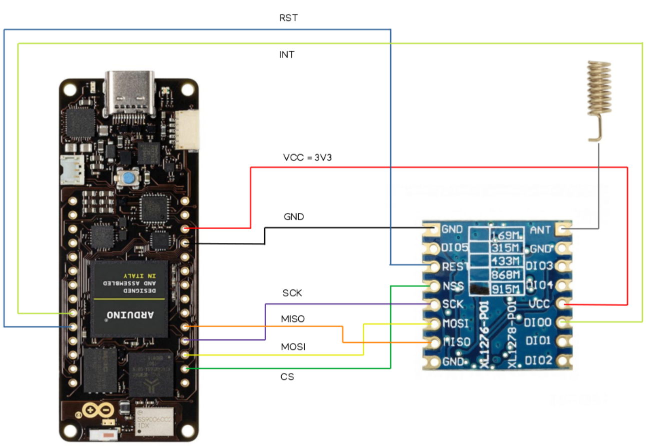

Figure 1. Connection for SX1276 chip to the Arduino Portenta H7.

III. Library compatibility

In order to validate the design, we first aim to verify that the new LoRa radio added to the Portenta supports the Arduino LoRa library[7]. We flash existing code fragments for LoRa sender and receiver to two Portenta H7 boards and could verify the successful communication between the two boards using the added LoRa radio.

The second test aims to validate that RadioLib can be used. With the additional LoRa radio board the SPI interface required by RadioLib is available. It needs to be mentioned that in order to use RadioLib, another interrupt signal has to be added by physically connecting the Portenta H7 with the DIO1 signal of the radio board, which finalizes the reception timer. Using a LoRa sender and receive code the communication between two Portenta boards over the added LoRa radio using RadioLib could be verified. An OLED display was used to verify the successful communication (Fig. 2).

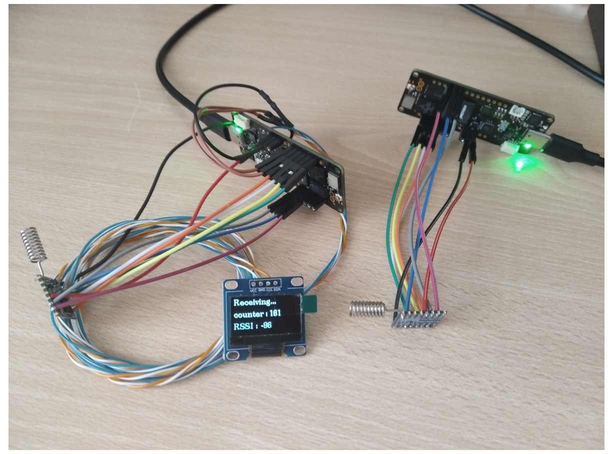

Figure 2. Communication test of two Arduino Portenta H7 by LoRa.

IV. Conclusions and outlook

We designed a double LoRa connectivity for the Arduino Portenta H7 to enable this board to become part of a LoRa mesh network. The new design extends the board's application to scenarios in which the device not only reaches the LoRaWAN gateway, but can also be part of a LoRa mesh network.

In future work we aim to integrate the new communication capacity into the Mbed OS available for the Arduino Portenta H7 and use it for interconnected machine learning applications.

Acknowledgment

This work was partially supported by the Spanish Government under contracts PID2019-106774RB-C21, PCI2019-111851-2 (LeadingEdge CHIST-ERA), PCI2019-111850-2 (DiPET CHIST-ERA).

References

[1] F. Sakr, F. Bellotti, R. Berta, and A. De Gloria, "Machine learning on mainstream microcontrollers," *Sensors*, vol. 20, no. 9, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/9/2638

[2] M. S. Yin and et al., "A lightweight deep learning approach to mosquito classification from wingbeat sounds," in *Proceedings of the Conference on Information Technology for Social Good*, ser. GoodIT '21, p. 37–42.

[3] M. A. M. Almuhaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, "A survey on lorawan technology: Recent trends, opportunities, simulation tools and future directions," *Electronics*, vol. 11, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/1/164

[4] R. P. Centelles, F. Freitag, R. Meseguer, and L. Navarro, "Beyond the star of stars: An introduction to multihop and mesh for LoRa and LoRaWAN," *IEEE Pervasive Computing*, vol. 20, no. 2, pp. 63–72, 2021.

[5] J. R. Cotrim and J. H. Kleinschmidt, "LoRaWAN mesh networks: A review and classification of multihop communication," *Sensors*, vol. 20, no. 15, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4273

[6] "Meshtastic: Open source hiking, pilot, skiing and secure GPS mesh communicator," https://meshtastic.org/, accessed: 2022-06-19.

[5] https://github.com/arduino/mkrwan1300-fw/issues/17

[6] https://github.com/GrumpyOldPizza/ArduinoCore-stm32l0

[7] https://github.com/sandeepmistry/arduino-LoRa