A **network** is a collection of interfaces that share a physical medium and a data protocol. A network allows software tasks in one computer to communicate and synchronize with software tasks running on another computer. For an embedded system, the network provides a means for distributed computing. The **topology** of a network defines how the components are interconnected. Examples topologies include rings, buses and multi-hop. Figure 11.10 shows a **ring** network of three microcontrollers. The advantage of this ring network is low cost and can be implemented on any microcontroller with a serial port. Notice that the microcontrollers need not be the same type or speed.

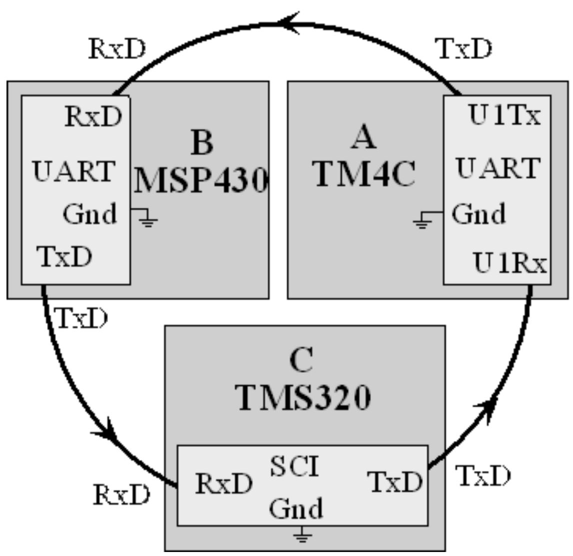

Figure 11.10. A simple ring network with three nodes, linked using the serial ports.

In this chapter we presented the hardware and software interfaces for the UART channel. We connected the TM4C to an I/O device and used the UART to communicate with the human. In this section, we will build on those ideas and introduce the concepts of networks by investigating a couple of simple networks. In particular, we will use the UART channel to connect multiple microcontrollers together, creating a network. A communication network includes both the physical channel (hardware) and the logical procedures (software) that allow users or software processes to communicate with each other. The network provides the transfer of information as well as the mechanisms for process synchronization.

When faced with a complex problem, one could develop a solution on one powerful and **centralized** computer system. Alternatively a **distributed** solution could be employed using multiple computers connected by a network. The processing elements in Figure 11.11 may be a powerful computer, a microcontroller, an application-specific integrated circuit (ASIC), or a smart sensor/actuator.

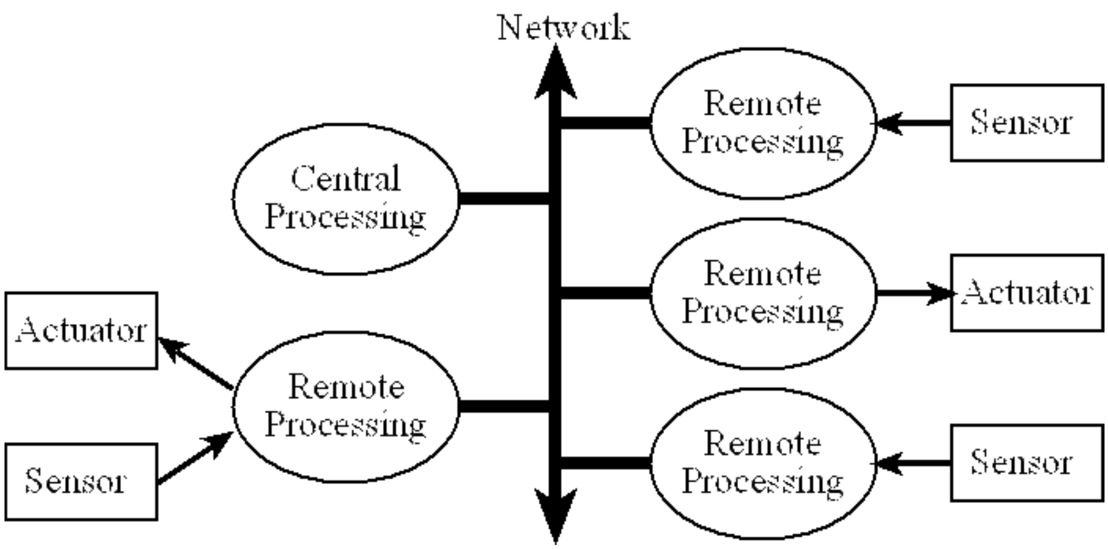

Figure 11.11. Distributed processing places input, output and processing at multiple locations connected together with a network.

There are many reasons to consider a distributed solution (network) over a centralized solution. Often multiple simple microcontrollers can provide a higher performance at lower cost compared to one computer powerful enough to run the entire system. Some embedded applications require input/output activities that are physically distributed. For real-time operation there may not be enough time to allow communication a remote sensor and a central computer. Another advantage of distributed system is improved debugging. For example, we could use one node in a network to monitor and debug the others. Often, we do not know the level of complexity of our problem at design time. Similarly, over time the complexity may increase or decrease. A distributed system can often be deployed that can be scaled. For example, as the complexity increases more nodes can be added, and if the complexity were to decrease nodes could be removed.

Help

Contact (https://www.edx.org/contact)

EdX is a non-profit created by founding partners Harvard and MIT whose mission is to bring the best of higher education to students of all ages anywhere in the world, wherever there is Internet access. EdX's free online MOOCs are interactive and subjects include computer science, public health, and artificial intelligence.

(http://www.meetup.com/edX-Global-Community/)

(http://www.facebook.com/EdxOnline)

(https://twitter.com/edXOnline)

(https://plus.google.com/108235383044095082735/posts)

(http://youtube.com/user/edxonline)

Terms of Service and Honor Code - Privacy Policy (https://www.edx.org/edx-privacy-policy)

Help