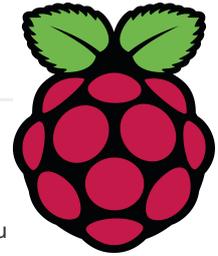


Activity: checksum



How to compute a checksum

Imagine sending the message "HELLO" as ASCII values. To calculate the checksum for this transmission, you add the ASCII values of all byte together. Then you convert this sum into ASCII, and send it along with the ASCII values for the letters of your message.

At the receiving end the checksum is computed from the received data and compared with the one you sent. If the two checksums, the data is assumed to be error-free. If the checksums do not match, the receivers know s there is an error somew here and discards the received data.

	ASCII value	Transmitted	-	Received	Computed
H	72	72	-	72	H
E	69	69	-	69	E
L	76	76	-	76	L
L	76	76	-	76	L
O	79	79	-	79	O
FCS		372	-		
FCS			-	372	

	ASCII value	Transmitted	-	Received	Computed
H	72	72		72	H
E	69	69	-	69	E
L	76	76	-	77 error	M
L	76	76	-	76	L
O	79	79	-	79	O
FCS		372	-		
FCS			-	373	

Classroom scenario

We want to send a word through a communications channel. Each letter of the word must first be converted into its ASCII code and then each ASCII code converted into binary.

In transit, a binary 1 may be regenerated as a binary 0 and vice versa.

Can you guarantee that the data you receive is the same as that which was sent?

Activity

1. Split the class up into teams of six, with two senders, two receivers, and two runners. Senders and receivers may not communicate with each other directly in any way.
2. Give the senders in each team a word. Have them choose who will convert to ASCII and who will convert to binary.
3. The first sender should convert each letter into ASCII, and pass it on to the second sender.
4. The second sender should convert each ASCII code to binary and write each 8-bit code on a piece of paper.
5. As each letter is fully encoded, one runner takes it to the teacher.
6. The teacher may change one of the bits to introduce an error, and passes each piece of paper on to the other runner.
7. The first receiver converts each 8-bit code to ASCII.
8. The second receiver converts the ASCII code back to a letter, and forms the word. Can the receivers tell if an error was introduced?

Then, repeat the exercise, but have the senders and their runner compute a checksum by:

1. Adding all the binary values together
2. Converting this checksum to binary
3. Sending it as a 16-bit final value

Finally, the receiving end should compute their own checksum to spot whether any errors have been introduced.

Tip: to really test your learners, use combinations of characters like "CLAS@".

Further activity

Investigate how we can use parity in a grid system to detect and correct errors, with [this free activity sheet](#)