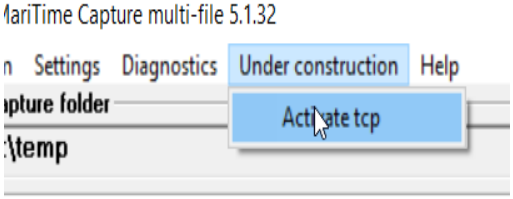
Testing the MariTime remote client

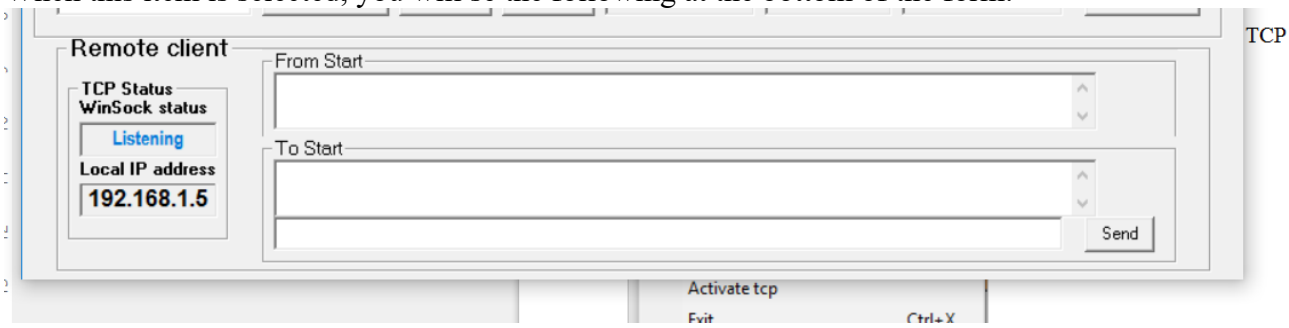
The *MariTime remote client* has been constructed in an attempt to enhance the functionality of MariTime in race conditions.
The remote client is assumed to be placed at the start, and it will cooperate with the MariTime program in the goal house.
Tecnically, the remote client and the central MariTime will communicate using TCP data communications. Normally this will probably means that a wireless IP link will be established between the two, and a LAN will be established. However there is no actual limitation to a LAN, and in reality the remote client and MariTime can communicate over any Internet connection (given that it is configured correctly)

The present remote client (and also the functions in MariTime) must be seen as *experimental*, and based on user feedback there will probably be changes over time.

On the **Video capture** form there is now a new menu item, **Activate TCP**



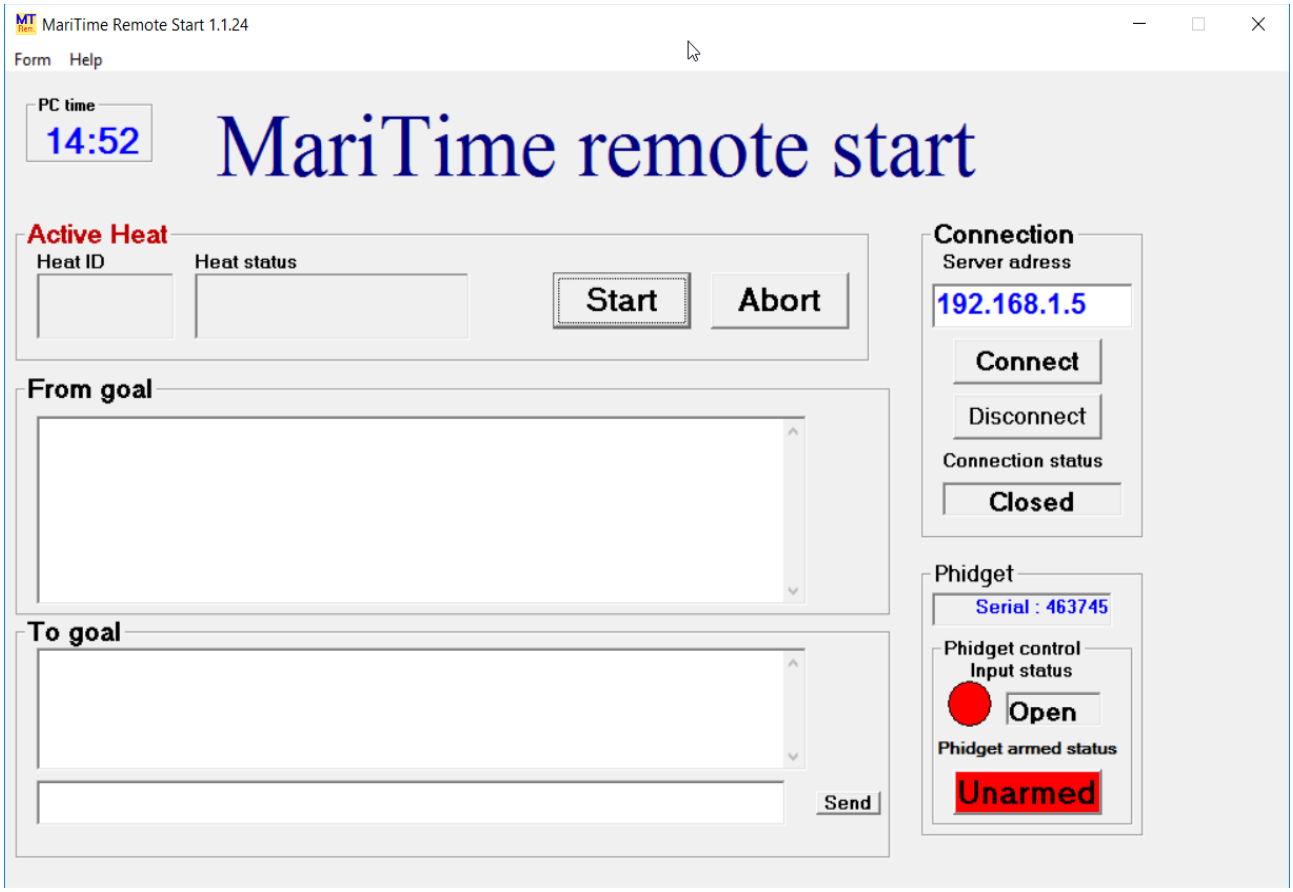
When this item is selected, you will se the following at the bottom of the form:



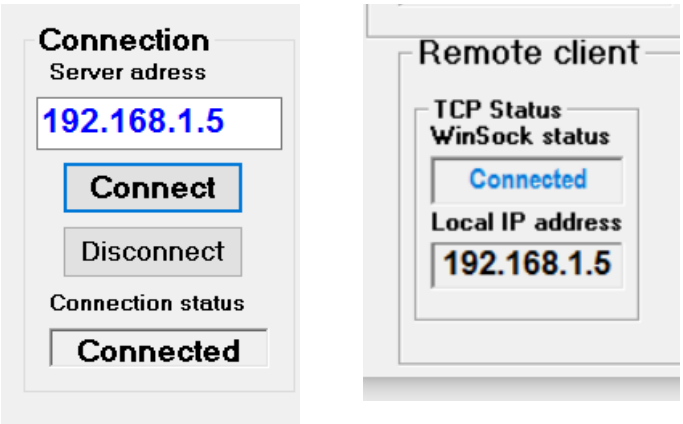
This frame is used to communicate with the remote client, as shown later.
In the TCP status frame there are two fields:
The **Winsock Status** will display the status of the communication with the remote client. Since this has not yet been started, we are just listening for an eventual connection request.
The Local IP address is important!
This is the address of MariTime, and this address must be known to, and used by the client to initiate the communication.
N.B. all communications is intended to be initiated/and terminated from the remote client. It is probably a good thing to allocate a "fixed address" to MariTime, this to avoid getting a variable address from a DHCP server. The communications uses port 6002
It is assumed that the remote client will be used inside a LAN (with radio link). However it is fully possible (I have tested it...) to have the remote client and the central MariTime communicate over any distance on internet. This will imply applying some internet communications skill outside of this document. And: my tests so far tell me that there is no noticeable delay for the start signal when on a LAN. Using internet, there will be some delay!

Starting the Remote Client.

In actual race conditions, the Remote client will of course be remote on a separate computer, but for initial testing, it is quite possible to have the two programs running in the same PC.
When the Remote client is started we see this form :

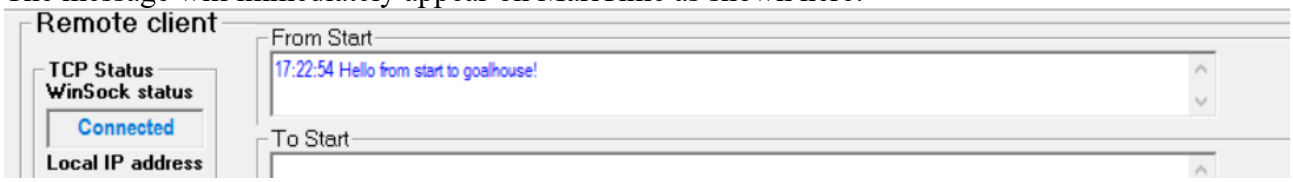


The server address in this example already contains the address of our server, and this is because the program "remembers" the address used earlier.
So, let us click **Connect** on the client!
If all goes well we will see this display on the client and the server



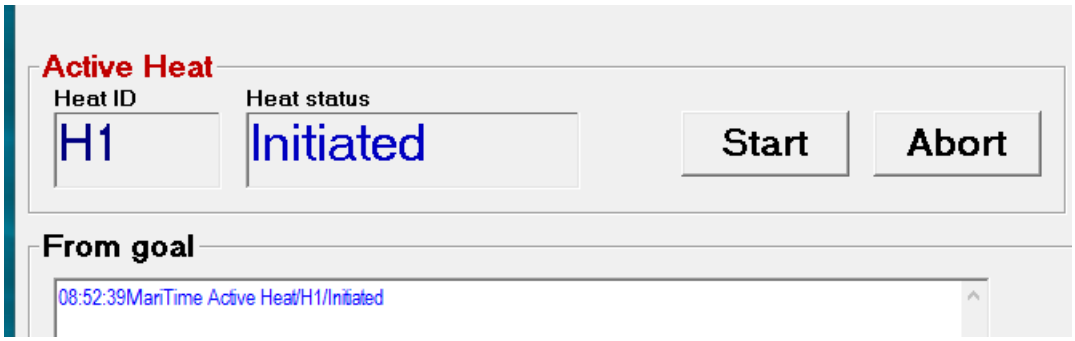
Now, we can start to communicate:
We can type something in the Remote client and click Send

The message will immediately appear on MariTime as shown here:



You should experiment with writing messages between the two programs. At present, the last 20 messages will be stored in a buffer, and you can scroll between them.
If a message contains the character "¤" (shift-4) it will trigger a distinctive "ring" sound at the receiving end!

Assuming that you have established a connection between he two programs, you can now try and initiate a file.
When you do this, you should see that the information about the new file appears on the Remote client like this:

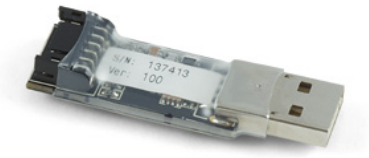


And, if you now click **Start** on the client, you will notice that the heat is actually started on MariTime!
Clicking **Abort** on the client will abort the heat, and clicking **Start** (presumably following a false start and a re-start) the heat will be started again.
Capturing will of course be done on MariTime in the normal way. The only actions that can be initiated from the client are **Start & Abort**.

If you initiate several files on MariTime and change the selected file between them you will notice that the client is always updated when the selection changes. The client will display the heat that is selected for *Start* (and not for capture, since this is not relevant to the client..).

Interfacing to start gates

We will use a simple Phidget hardware like this:



This is actually a “little brother” of the Phidget board that is used in the Control boxes.
This one plugs directly into a USB board on the client PC and the two wires from the start equipment are connect directly to it.
The wires to connect are the black & Yellow.

If no Phidget is connected, the Phidget frame looks like this:

Once we connect a Phidget to a USB port, the display should change to this:

We can see the serial number of the Phidget. We can also see the status of the digital input (black&yellow) that we use.

At present there is no connection between the two, but if we connect them we should notice a change to this:

The **Phidget armed status** is a button that when clicked will alternate between ***Unarmed*** and *Armed*.

When Unarmed, the program will display the current input status, but nothing else will happen.

However, when the status is *Armed* any input change will result in the program sending a Start signal. The function will be the same as clicking the Start button.

The program does not care in the input goes from open to closed ot the other way around.
But it is recommended that the starting gates are wired so that they give a closed connection waiting for start, and that the contact opens on start.

Important information on Phidget drivers

The Phidget hardware used in the Maritime system works using special software (drivers) that are installed together with the MariTime program.

The Control boxes use drivers version 2.1.6.

However, the new Phidget used by the remote client uses drivers 2.1.8 (it must have them)

But, if you try to run MariTime (expecting drivers 2.1.6) on a PC with version 2.1.8 installed, there may be trouble.

Therefore, for the time being: Only run the remote client on a dedicated PC, and do not install it in a PC intended for use with MariTime!