syn1588 ®

Application Note AN009

GPS Receivers for syn1588® NICs

Version 1.14.0 - January 2022

Abstract

The syn1588® PCIe NIC as well as the syn1588® Dual NIC allow the connection of an external GPS receiver that provides the absolute time for IEEE1588 master operations. This application note briefly describes the basic setup procedure as well as the requirements for selecting an appropriate GPS receiver.

Using a GPS Receiver

Using an external GPS receiver to supply an absolute time reference for a syn1588® NIC is a simple task. One just requires two connections:

- the 1PPS signal
- the serial NMEA data stream

The following figure shows the basic connection scenario when connecting a syn1588® NIC with a GPS receiver. Please note, while basically there is no difference between the basic functions of the syn1588® PCIe NIC and the syn1588® Dual NIC there are some specific limitations for the syn1588® Dual NIC due to the limited number of SMA IOs. Please check the Application Note AN0036 "Special considerations when using IOs of the syn1588® Dual NIC" for further details.

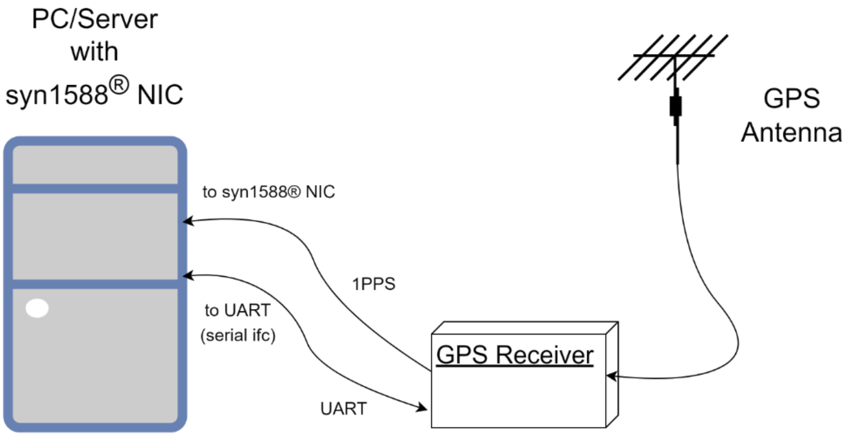

Figure 1 Basic connection scenario for a GPS receiver

The 1PPS signal provides the exact phase information while the serial data stream provides the absolute time information for each 1PPS event. The GPS receiver has to support the NMEA-0183 protocol (including the RMC messages).

The following NMEA output example shows only a small part of the possible log files and additional messages. NMEA has its own version of essential gps pvt (position, velocity, time) data. It is called RMC.

```
$GPRMC, 123519,A,4807.038,N,01131.000,E,022.4,084.4,230317,003.1,W,F*6A
```

The data needed by eSync are in this example:

123519	= time of day	12:35:19 UTC
230317	= date of day	23 of march 2017

If the seconds in time of day have the value 60 a leap second will be applied i.e. the utcOffset will be increased by 1.

From the software point of view there is the syn1588® PTP Stack required as well as the eSync utility. Both software come with every syn1588® NIC.

The basic sequence of invoking the software is:

- invoke the syn1588® PTP Stack first instructing to act as PTP Master
- wait some seconds
- invoke eSync informing which SMA IO connector and UART shall be used for communication

Naturally, this sequence can be scripted to simply things. For a detailed explanation of both eSync and the syn1588® PTP Stack please check the syn1588® User Guide. The following eSync invocation shows a typical example:

```
sudo ./eSync -p 4 -c ttyUSB0 -b 4800 -v2 -s -d 0
```

The command line options used specify the following behavior:

- -p 4: use connector X4 for 1PPS signal
- -c ttyUSB0: use this (Linux) device for the serial interface
- -b 4800: select a baud rate of 4800
- -v2: set a verbosity level of 2 on the output
- -s: use shared memory to control/instruct the syn1588® PTP Stack
- -d 0: use card 0

A typical command output may look like this:

```
oregano@KAEFER:/opt/oregano/bin$ sudo ./eSync -p 7 -c ttyUSB0 -b 4800 -v info -d 0
2022-01-05 12:46:09.692868 [INFO    ] [              ]  syn1588(R) eSync - External
Synchronization Engine
2022-01-05 12:46:09.692969 [INFO    ] [              ]  Build date: 2022-01-
04T10:06:37 - v1.14-0-g6624ce7d-DIRTY
2022-01-05 12:46:09.692998 [INFO    ] [              ]  Copyright (c) Oregano
Systems - Design & Consulting GesmbH 2005-2022
2022-01-05 12:46:09.693019 [INFO    ] [              ]  Confidential unpublished
data - All rights reserved
2022-01-05 12:46:09.693055 [INFO    ] [              ]  Port 0: adding config "p" =
"7"
2022-01-05 12:46:09.693081 [INFO    ] [              ]  Port 0: adding config "c" =
"ttyUSB0"
2022-01-05 12:46:09.693104 [INFO    ] [              ]  Port 0: adding config "b" =
"4800"
2022-01-05 12:46:09.693126 [INFO    ] [              ]  Port 0: adding config "v" =
"info"
2022-01-05 12:46:09.693147 [INFO    ] [              ]  Port 0: adding config "d" =
"0"
2022-01-05 12:46:09.693216 [WARNING ] [              ]  Please note that Oregano
Systems' eSync utility is not SyncE - Synchronous Ethernet!
2022-01-05 12:46:09.693244 [INFO    ] [syn1588       ]  Syn1588Ifc requires at
least:
2022-01-05 12:46:09.693264 [INFO    ] [syn1588       ]  - linux driver version 1.4-
15-g05b7283
2022-01-05 12:46:09.693288 [INFO    ] [syn1588       ]  - windows driver version
10/05/2017, 10.9.16.182
2022-01-05 12:46:09.693353 [INFO    ] [syn1588       ]  Device /dev/syncD0 found
2022-01-05 12:46:09.693408 [INFO    ] [syn1588       ]  syn1588(R) Hardware Clock M
2.3.5 f=125000000 Hz
2022-01-05 12:46:09.693441 [INFO    ] [syn1588       ]  Found stop clock support
2022-01-05 12:46:09.693473 [INFO    ] [syn1588       ]  Using MAC TS Version 3160
2022-01-05 12:46:09.693497 [INFO    ] [syn1588       ]  Using programmable 1-step
TS
2022-01-05 12:46:09.693544 [INFO    ] [syn1588       ]  syn1588(R) PCIe NIC
Revision 2, Build 865
2022-01-05 12:46:09.693627 [INFO    ] [clock         ]  Spike M2S: Init with ival
0, buffer size 16
2022-01-05 12:46:09.693904 [INFO    ] [clock         ]  Spike Path: Init with ival
0, buffer size 16
2022-01-05 12:46:09.693989 [INFO    ] [clock         ]  Clk: Using Oregano Systems;
syn1588(R) PCIe NIC Revision 2; 00:1E:C0:85:DE:2B
2022-01-05 12:46:09.694015 [INFO    ] [clock         ]    with ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 12:46:09.694051 [INFO    ] [clock         ]  Clk: Resetting servos
2022-01-05 12:46:09.694071 [INFO    ] [clock         ]  Clk: Resetting filters
2022-01-05 12:46:09.694090 [INFO    ] [clock         ]  Spike M2S: Init with ival
0, buffer size 16
2022-01-05 12:46:09.694114 [INFO    ] [clock         ]  Spike Path: Init with ival
0, buffer size 16
2022-01-05 12:46:09.968472 [INFO    ] [              ]  Using shared memory
2022-01-05 12:46:09.968559 [INFO    ] [io            ]  Init shared mem
2022-01-05 12:46:09.968743 [INFO    ] [CeSyncengine ]  Using Nmea and PPS for
synchronization
```

After the information header from the eSync utility the synchronization process will deliver also information on the process of synchronization depending on the verbosity level. Below, a typical synchronization process output with verbosity level info is shown.

```
2022-01-05 12:46:22.001936 [INFO    ] [clock        ]  Update M2S-Delay -60 ns
2022-01-05 12:46:22.001963 [INFO    ] [clock        ]  DriftCalc: drift is 0 ns/s
2022-01-05 12:46:22.002031 [INFO    ] [syn1588      ]  Adjust rate by -4195.35
ns/s (speeding up)
2022-01-05 12:46:22.002063 [INFO    ] [syn1588      ]  Setting step to 0x00000800,
0x023316ff, 8.000033563 ns
2022-01-05 12:46:22.002112 [INFO    ] [clock        ]  Clk: Resetting filters
2022-01-05 12:46:22.002138 [INFO    ] [clock        ]  Spike M2S: Init with ival
0, buffer size 16
2022-01-05 12:46:22.002163 [INFO    ] [clock        ]  Spike Path: Init with ival
0, buffer size 16
2022-01-05 12:46:22.002206 [INFO    ] [clock        ]  Drift calc completed
2022-01-05 12:46:23.001886 [INFO    ] [CeSyncengine ]  UTCTime: 114622 UTCDate:
050122
2022-01-05 12:46:23.001948 [INFO    ] [clock        ]  Update M2S-Delay -65 ns
2022-01-05 12:46:24.001539 [INFO    ] [CeSyncengine ]  UTCTime: 114623 UTCDate:
050122
2022-01-05 12:46:24.001574 [INFO    ] [clock        ]  Update M2S-Delay -69 ns
2022-01-05 12:46:25.001883 [INFO    ] [CeSyncengine ]  UTCTime: 114624 UTCDate:
050122
2022-01-05 12:46:25.001942 [INFO    ] [clock        ]  Update M2S-Delay -66 ns
2022-01-05 12:46:25.001970 [INFO    ] [syn1588      ]  Adjust rate by -4261.35
ns/s (speeding up)
2022-01-05 12:46:25.001997 [INFO    ] [syn1588      ]  Setting step to 0x00000800,
0x023bf2be, 8.000034091 ns
2022-01-05 12:46:26.001539 [INFO    ] [CeSyncengine ]  UTCTime: 114625 UTCDate:
050122
2022-01-05 12:46:26.001561 [INFO    ] [clock        ]  Update M2S-Delay -5 ns
2022-01-05 12:46:26.001568 [INFO    ] [clock        ]  Spike M2S: Discarding -5
```

First the drift and actual offset is calculated. Afterwards the control loop starts to lock

```
2022-01-05 12:46:26.001598 [INFO    ] [clock        ]  Clk: Resetting filters
2022-01-05 12:46:26.001604 [INFO    ] [clock        ]  Spike M2S: Init with ival
0, buffer size 16
2022-01-05 12:46:26.001610 [INFO    ] [clock        ]  Spike Path: Init with ival
0, buffer size 16
2022-01-05 12:46:26.001616 [INFO    ] [clock        ]  Clk: slewed clock.
2022-01-05 12:46:26.001621 [INFO    ] [clock        ]  changing state to
EFastFiltering
2022-01-05 12:46:27.002055 [INFO    ] [CeSyncengine ]  UTCTime: 114626 UTCDate:
050122
2022-01-05 12:46:27.002146 [INFO    ] [clock        ]  Update M2S-Delay -9 ns
2022-01-05 12:46:28.001859 [INFO    ] [CeSyncengine ]  UTCTime: 114627 UTCDate:
050122
2022-01-05 12:46:28.001924 [INFO    ] [clock        ]  Update M2S-Delay -6 ns
2022-01-05 12:46:29.001863 [INFO    ] [CeSyncengine ]  UTCTime: 114628 UTCDate:
050122
2022-01-05 12:46:29.001927 [INFO    ] [clock        ]  Update M2S-Delay -11 ns
2022-01-05 12:46:29.001962 [INFO    ] [clock        ]  Adjusting clock at -11.00
ns offset
…
```

Selecting a GPS Receiver

For connecting a GPS receiver to either a syn1588® PCIe NIC or a syn1588® Dual NIC one requires two connections:

- the 1PPS signal
- the serial NMEA data stream

While the 1PPS signal is connected to any of the available User I/Os (SMA connectors) of the syn1588® NIC any standard UART interface of the PC or server – even USB-to-Serial converters or cables will do – can be utilized for connecting the serial line. Basically, almost every GPS receiver provides both a 1 PPS signal as well as a serial NMEA0183 stream (via UART).

Please check the signaling of the 1PPS signal provided by the GPS receiver against the specifications of the syn1588® PCIe NIC or a syn1588® Dual NIC. The syn1588® NICs expect a 50 ohm 3V3 signal. Please check the respective datasheet for a full specification of the interface.

Please note, that when connecting a GPS receiver to a syn1588® PCIe NIC one has to operate the syn1588® card in IEEE1588 Master mode. Please further note, that we generally recommend the high-stability OCXO oscillator option for all IEEE1588 Master applications.

Meinberg GPS Solution

If one requires a reliable and accurate solution, Oregano Systems recommends GPS receivers from Meinberg (http://www.meinberg.de). For example the

Meinberg GPS180

(http://www.meinbergglobal.com/english/products/3u-gps-clock-lc-display.htm).

Figure 2 Meinberg GPS180

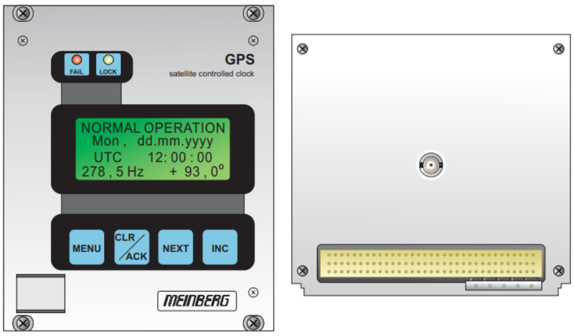

Figure 3 Meinberg GPS180 (front/rear view basic configuration)

Required Equipment for Meinberg GPS180

Connector: 96-pin VG-female DIN 41612 a+b+c

Ordering @ Digikey: manufacturer: Harting, part number: 09032966845

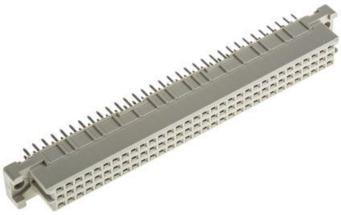

Figure 4 96-pin VG-female DIN 41612 a+b+c

Pin	Type
1	VCC in (+5V)
2	VCC in (+12V)
3	
4	PPS out
5	GND
6	PPS in
7	GND
8	TC_DCLS in
9	TC_AM in
10	Reserve 0
11	GND
12	-4.096MHz in
13	+4.096MHz in
14	GND
15	Board_ID0
16	Board_ID1
17	Board_ID2
18	Board_ID3
19	Time Sync in
20	GND
21	10MHz in
22	GND
23	Reserve 1
24	RxD in
25	Slot_ID0
26	Slot_ID1
27	Slot_ID2
28	Slot_ID3
29	+USB

30	-USB
31	GND
32	GND

Table 1Pin assignment GPS180

Attenuator: 3db for 1PPS

Figure 5 Attenuator 3DB 50Ohm SMA

Ordering @ Digikey: Manufacturer: Crystek Corporation, part number: CATTEN-03R0

Cable: USB-A to open end

Figure 6 Cable USB-A to open end

Ordering @ Digikey: Manufacturer: Molex, part number: 0887283200

Test setup with Meinberg GPS180

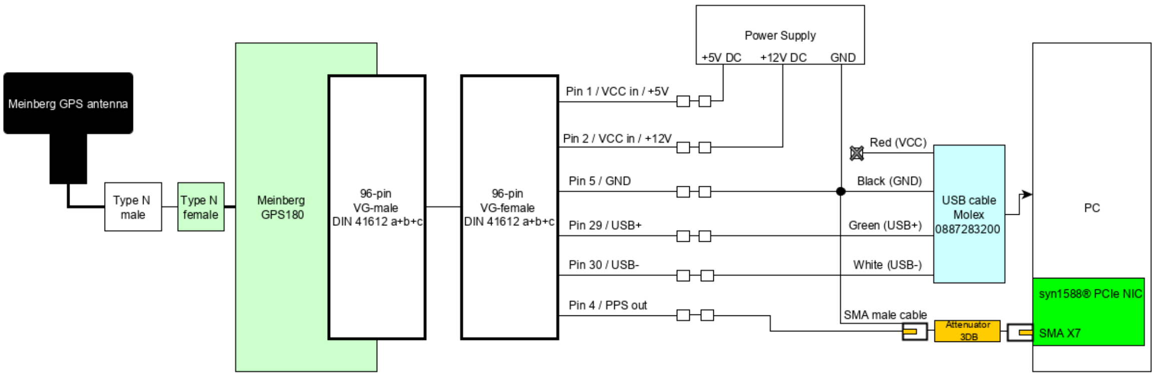

Figure 7 Meinberg Test setup

Navilock GPS Solution

If one requires a simple device for testing, a Navilock receiver (https://www.navilock.com/) is fine as well. For example the

Navilock NL-8004P

(https://www.navilock.com/produkte/G_62527/merkmale.html).

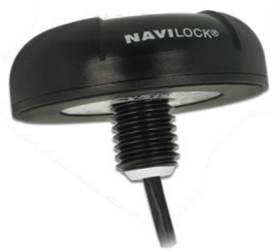

Figure 8 Navilock NL-8004P

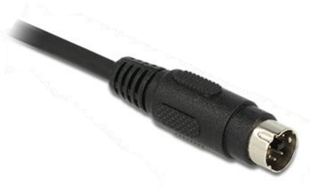

Figure 9 Navilock Connector MD6 male

Required Equipment for Navilock NL-8004P

Cable: Navilock 62928

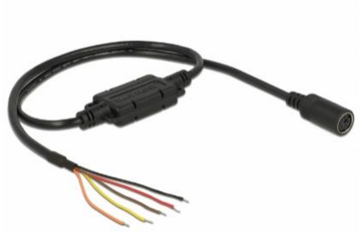

Figure 10 Navilock MD6 female serial> 5 x open wire LVTTL (3.3 V) 52 cm cable

Ordering @ Navilock: part number: 62928

Cable: FTDI TTL-232R-3V3-WE

Figure 11 FTDI TTL-232R-3V3-WE cable

Ordering @ Digikey: Manufacturer: FTDI, part number: TTL-232R-3V3-WE

Cable: SMA male to open end

Figure 12 SMA male cable

Resistor: 10k Ohm

Figure 13 Resistor 1k2 Ohm

Test setup with Navilock NL-8004P

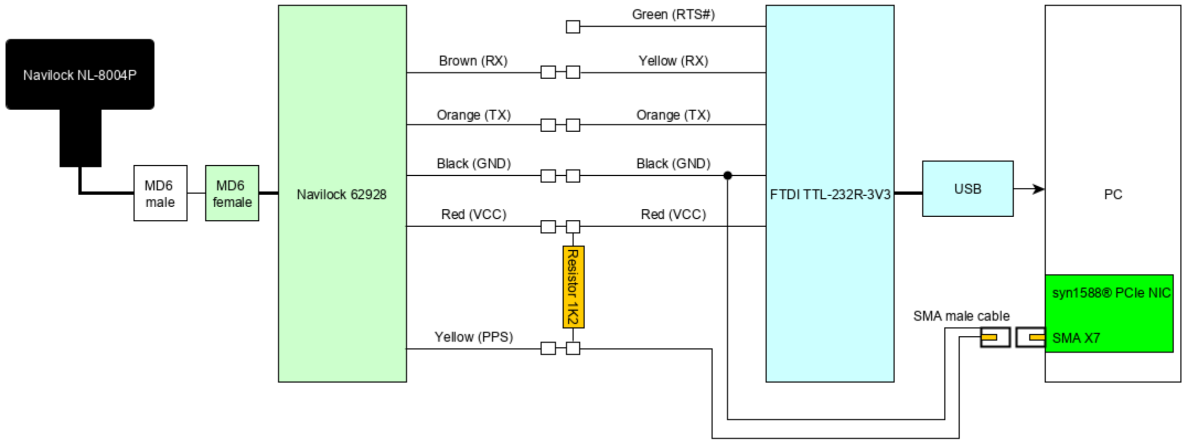

Figure 14 Navilock Test setup

Example eSync output

This example shows the eSync console output while synchronizing the syn1588® PCIe NIC to an Navilock NL-8004P GPS receiver by using an 1 PPS signal and the serial NMEA data stream. For detailed information on how to use eSync please refer to our syn1588® User Guide chapter 6.2 “eSync - Synchronizing to External Sources”

```
C:\Users\admin\Desktop\ptpmmmGUI_build_v1.8-17-g0e6d01ad\sw\apps\windows>esync.exe
-c COM7 -b 9600 -p1 -v2
syn1588(R) eSync - External Synchronization Engine
Build date: Aug  5 2019 - V 1.9-12 Rev gf5928642
Copyright (c) Oregano Systems - Design & Consulting GesmbH 2005-2018
Confidential unpublished data - All rights reserved

Please note that Oregano Systems' eSync utility is not SyncE - Synchronous
Ethernet

Command line: esync.exe -c COM7 -b 9600 -p1 -v2
Port 0: adding config "c" = "COM7"
Port 0: adding config "b" = "9600"
Port 0: adding config "p" = "1"
Port 0: adding config "v" = "2"
Syn1588Ifc requires at least:
- linux driver version 1.4-15-g05b7283
- windows driver version 10/05/2017, 10.9.16.182
syn1588(R) Hardware Clock M 2.3.3 f=125000000 Hz
syn1588(R) PCIe NIC Revision 2, Build 825 with HQ Oscillator
Clk: Using Oregano Systems; syn1588(R) PCIe NIC Revision 2; 00:1E:C0:85:D0:FD
  with ClockId 00:1E:C0:FF:FE:85:D0:FD
Clk: Resetting servos
Clk: Resetting filters
Using Nmea and PPS for synchronization
Warning: RMC message not detected.
Warning: No valid reference time received
Using external synchronization source with reference time
Starting synchronization process
Warning: RMC message not detected.
Warning: No valid reference time received
Update M2S-Delay -3 ns
Starting slew lock
Update M2S-Delay -1000000010 ns
Clk: Timestamp out of range: 1569855271.000000032
Update M2S-Delay -1000000008 ns
Clk: Timestamp out of range: 1569855272.000000032
Update M2S-Delay -999999982 ns
Clk: Setting time to Origin : 1569855273.000000032
Clk: Resetting servos
Clk: Resetting filters
Warning: RMC message not detected.
Warning: No valid reference time received
Update M2S-Delay -260464769 ns
Update M2S-Delay 739535319 ns
Clk: Timestamp out of range: 1569855274.000000032
Update M2S-Delay 739535383 ns
Clk: Timestamp out of range: 1569855275.000000032
Update M2S-Delay 739535471 ns
Clk: Setting time to Origin : 1569855276.000000032
Clk: Resetting servos
Clk: Resetting filters
Update M2S-Delay -5544280 ns
Update M2S-Delay -5544200 ns
Update M2S-Delay -5544112 ns
Update M2S-Delay -5544040 ns
Update M2S-Delay -5543960 ns
Update M2S-Delay -5543888 ns
Update M2S-Delay -5543800 ns
Update M2S-Delay -5543736 ns
Update M2S-Delay -5543648 ns
DriftCalc: drift is 81 ns/s
Clk: Resetting filters
Drift calc completed
Update M2S-Delay -5543665 ns
Update M2S-Delay -5543658 ns
Update M2S-Delay -5543651 ns
Update M2S-Delay -18728 ns
Clk: Resetting filters
Clk: slewed clock.
changing state to EFastFiltering
Update M2S-Delay 100 ns
Update M2S-Delay 83 ns
Update M2S-Delay 90 ns
Adjusting clock at 90.00 ns offset
Update M2S-Delay 49 ns
Adjusting clock at 19.00 ns offset
Update M2S-Delay -7 ns
Adjusting clock at -7.00 ns offset
Update M2S-Delay -1 ns
Adjusting clock at -1.00 ns offset
Update M2S-Delay -6 ns
Adjusting clock at -6.00 ns offset
Clk: Sync in bound      : 1569855300.005156207 Ofs:      -6 ns
Update M2S-Delay -12 ns
Adjusting clock at -12.00 ns offset
Update M2S-Delay -7 ns
```

Example eSync output

Version 1.14.0 - January 2022

Oregano Systems
A Meinberg Company

Franzosengraben 8
A-1030 Vienna
Austria
http://oregano.at
contact@oregano.at

Oregano Systems does not assume any liability arising out of the application or use of any product described or shown herein nor does it convey any license under its patents, copyrights, or any rights of others.

Licenses or any other rights such as, but not limited to, patents, utility models, trademarks or tradenames, are neither granted nor conveyed by this document, nor does this document constitute any obligation of the disclosing party to grant or convey such rights to the receiving party.

Oregano Systems reserves the right to make changes, at any time without notice, in order to improve reliability, function or design. Oregano Systems will not assume responsibility for the use of any circuitry described herein.

All trademarks used in this document are the property of their respective owners.