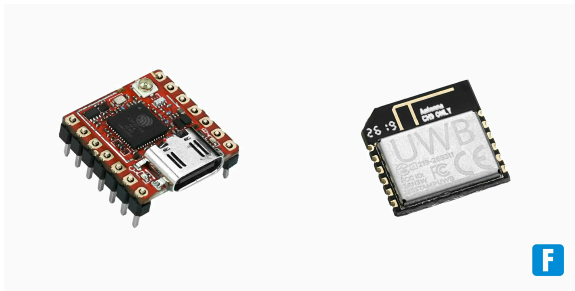


Stamp UWB F Arduino Tutorial

1. Preparation

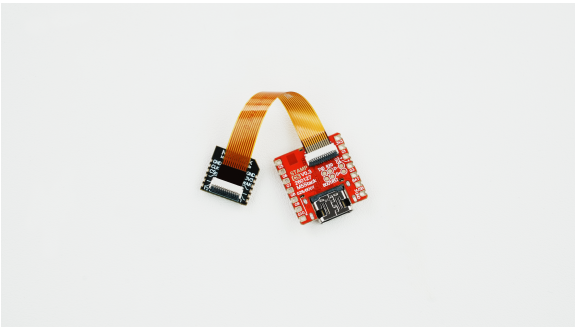
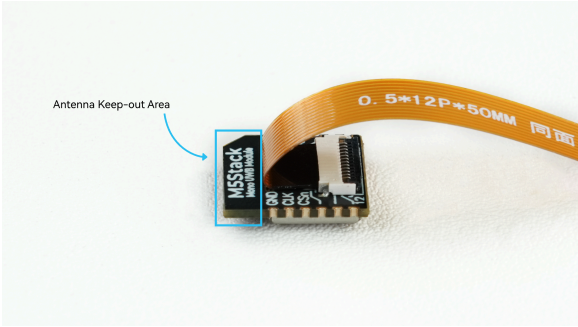
- Environment Setup: Refer to the [Arduino IDE Getting Started Guide](#) to complete the IDE installation, and install the board manager and drivers for the Stamp C5.
- Libraries Used:
 - [M5Stamp-UWB](#)
- Hardware Products Used:
 - [Stamp-C5 DIP](#)
 - [Stamp UWB F](#)



2. Notes

Note

1. To ensure stable communication, power the Stamp UWB with a high-PSRR LDO or a power supply with ripple below 12mVpp.
2. UWB ranging requires at least two devices: one device acting as the Tag and the other as the Anchor.
3. When designing the complete device, consider the impact of the FPC cable and its connector on the module antenna performance. Do not route the FPC cable beneath the module's PCB antenna or through the antenna keep-out area, as this may alter the antenna impedance and radiation characteristics, degrade RF performance, and introduce ranging errors. Route the FPC cable outside the antenna area, and test the fully assembled device to ensure that its communication range and ranging accuracy meet the design requirements.



Pin Compatibility

Stamp UWB communicates over SPI. The examples below use Stamp C5 as the host; if you use a different host or carrier board, adjust the pin definitions in the code accordingly.

The pin connections between Stamp C5 and Stamp UWB F are as follows:

Stamp UWB F Pin	Stamp C5 GPIO
GP7	G23
IRQ	G0
WAKEUP	G24
RST	G25
MISO	G26
MOSI	G27
CS	G11
SCK	G12

3. Double-Sided Two-Way Ranging (DS-TWR)

3.1 Tag

```
cpp
```

```

1  #include <Arduino.h>
2  #include <M5Stamp_UWB.h>
3
4  static constexpr uint32_t RANGE_INTERVAL_MS = 200;
5  static constexpr uint32_t LOG_INTERVAL      = 10;
6
7  M5Stamp_UWB uwb;
8  M5Stamp_UWBDSRangeConfig rangeConfig;
9  bool uwbReady          = false;
10 uint32_t lastRangeMs = 0;
11 uint32_t rangeCount   = 0;
12 uint32_t okCount      = 0;
13
14 static bool initUwb()
15 {
16     // Stamp-C5 connections to the onboard QM33120 UWB transceiver.
17     M5Stamp_UWBConfig config;
18     config.pin_gp7      = 23;
19     config.pin_irq      = 0;
20     config.pin_wakeup   = 24;
21     config.pin_rst      = 25;
22     config.pin_miso     = 26;
23     config.pin_mosi     = 27;
24     config.pin_cs       = 11;
25     config.pin_sck      = 12;
26
27     M5Stamp_UWBPHYConfig phy;
28     phy.channel = M5Stamp_UWBChannel::Channel5;
29
30     // Both devices must use the same network addresses and DS-TWR timing.
31     rangeConfig.panId                = 0xDECA;
32     rangeConfig.initiatorAddress     = 0x0001;
33     rangeConfig.responderAddress     = 0x0002;
34     rangeConfig.responseRxAfterTxDelayUus = 1500;
35     rangeConfig.responseTxDelayUus   = 3000;
36     rangeConfig.finalTxDelayUus      = 1800;
37     rangeConfig.finalRxAfterResponseTxDelayUus = 500;
38     rangeConfig.resultRxAfterFinalTxDelayUus = 500;
39     rangeConfig.rxTimeoutUus         = 3000;
40     rangeConfig.hostTimeoutMs        = 100;
41     rangeConfig.resultRepeatCount    = 3;
42     rangeConfig.resultRepeatGapMs    = 3;
43
44     if (!uwb.begin(config, phy)) {
45         Serial.printf("UWB_BEGIN,result=FAIL,error=%s\n", uwb.lastErrorName());
46         Serial.printf("UWB_RAW_ID,dev_id=0x%08lX\n", static_cast<unsigned long>(uwb.readRawDeviceId()));
47         return false;
48     }
49
50     const uint32_t devId = uwb.deviceId();
51     Serial.printf("UWB_ID,dev_id=0x%08lX,chip=%s\n", static_cast<unsigned long>(devId), uwb.chipName());
52     if (devId != M5STAMP_UWB_QM33120_DEVICE_ID) {
53         Serial.printf("UWB_ID,result=FAIL,expected=0xDECA0314\n");
54         return false;
55     }

```

```

56
57     Serial.printf("UWB_CONFIG,result=OK,ch=%u,plen=%u,rate=6M8,tx_power=0x%08lX\n", static_cast<unsigned
58         static_cast<unsigned>(phy.preambleLength), static_cast<unsigned long>(phy.txPower));
59     return true;
60 }
61
62 static void runRanging()
63 {
64     // Start one DS-TWR exchange at the configured interval.
65     if (millis() - lastRangeMs < RANGE_INTERVAL_MS) return;
66     lastRangeMs = millis();
67
68     const M5Stamp_UWBDSRangeResult result = uwb.requestDSRange(rangeConfig);
69     rangeCount++;
70     okCount += result.success;
71
72     // Print accumulated statistics every LOG_INTERVAL attempts.
73     if (rangeCount % LOG_INTERVAL) return;
74
75     const uint32_t failCount = rangeCount - okCount;
76     if (result.success) {
77         Serial.printf(
78             "DS_RANGE_STAT,count=%lu,ok=%lu,fail=%lu,last=OK,seq=%u,distance_mm=%ld,distance_m=%.3f,elap
79             static_cast<unsigned long>(rangeCount), static_cast<unsigned long>(okCount),
80             static_cast<unsigned long>(failCount), result.sequence, static_cast<long>(result.distanceMm)
81             result.distanceM, static_cast<unsigned long>(result.elapsedMs));
82     } else {
83         Serial.printf("DS_RANGE_STAT,count=%lu,ok=%lu,fail=%lu,last=FAIL,seq=%u,error=%s\n",
84             static_cast<unsigned long>(rangeCount), static_cast<unsigned long>(okCount),
85             static_cast<unsigned long>(failCount), result.sequence, uwb.lastErrorName());
86     }
87 }
88
89 void setup()
90 {
91     Serial.begin(115200);
92     delay(1000);
93     Serial.printf("M5Stamp UWB DS-TWR TAG\n");
94     Serial.printf("ROLE,mode=TAG\n");
95     Serial.printf("TWR_MODE,mode=DS-TWR\n");
96     uwbReady = initUwb();
97     Serial.printf("TEST_START,result=%s\n", uwbReady ? "OK" : "FAIL");
98 }
99
100 void loop()
101 {
102     if (uwbReady) runRanging();
103     else delay(1000);
104 }

```

Serial output example:


```
M5Stamp UWB DS-TWR TAG
ROLE,mode=TAG
TWR_MODE,mode=DS-TWR
UWB_ID,dev_id=0xDECA0314,chip=QM33120/DW3720
UWB_CONFIG,result=OK,ch=5,plen=128,rate=6M8,tx_power=0xFEFEFEFE
TEST_START,result=OK
DS_RANGE_STAT,count=10,ok=10,fail=0,last=OK,seq=10,distance_mm=164,distance_m=0.164,elapsed_ms=7
DS_RANGE_STAT,count=20,ok=20,fail=0,last=OK,seq=20,distance_mm=166,distance_m=0.166,elapsed_ms=7
DS_RANGE_STAT,count=30,ok=30,fail=0,last=OK,seq=30,distance_mm=171,distance_m=0.171,elapsed_ms=7
DS_RANGE_STAT,count=40,ok=40,fail=0,last=OK,seq=40,distance_mm=168,distance_m=0.168,elapsed_ms=7
DS_RANGE_STAT,count=50,ok=50,fail=0,last=OK,seq=50,distance_mm=153,distance_m=0.153,elapsed_ms=7
DS_RANGE_STAT,count=60,ok=60,fail=0,last=OK,seq=60,distance_mm=196,distance_m=0.196,elapsed_ms=7
DS_RANGE_STAT,count=70,ok=70,fail=0,last=OK,seq=70,distance_mm=172,distance_m=0.172,elapsed_ms=7
DS_RANGE_STAT,count=80,ok=80,fail=0,last=OK,seq=80,distance_mm=182,distance_m=0.182,elapsed_ms=7
DS_RANGE_STAT,count=90,ok=90,fail=0,last=OK,seq=90,distance_mm=160,distance_m=0.160,elapsed_ms=7
DS_RANGE_STAT,count=100,ok=100,fail=0,last=OK,seq=100,distance_mm=157,distance_m=0.157,elapsed_ms=7
DS_RANGE_STAT,count=110,ok=110,fail=0,last=OK,seq=110,distance_mm=210,distance_m=0.210,elapsed_ms=7
DS_RANGE_STAT,count=120,ok=120,fail=0,last=OK,seq=120,distance_mm=164,distance_m=0.164,elapsed_ms=7
DS_RANGE_STAT,count=130,ok=130,fail=0,last=OK,seq=130,distance_mm=174,distance_m=0.174,elapsed_ms=7
```

3.2 Anchor

cpp

```

1  #include <Arduino.h>
2  #include <M5Stamp_UWB.h>
3
4  static constexpr uint32_t LOG_INTERVAL = 20;
5
6  M5Stamp_UWB uwb;
7  M5Stamp_UWBDSRangeConfig rangeConfig;
8  bool uwbReady          = false;
9  uint32_t responseCount = 0;
10 uint32_t failCount      = 0;
11
12 static bool initUwb()
13 {
14     // Stamp-C5 connections to the onboard QM33120 UWB transceiver.
15     M5Stamp_UWBConfig config;
16     config.pin_gp7      = 23;
17     config.pin_irq      = 0;
18     config.pin_wakeup   = 24;
19     config.pin_rst      = 25;
20     config.pin_miso     = 26;
21     config.pin_mosi     = 27;
22     config.pin_cs       = 11;
23     config.pin_sck      = 12;
24
25     M5Stamp_UWBPHYConfig phy;
26     phy.channel = M5Stamp_UWBChannel::Channel5;
27
28     // Both devices must use the same network addresses and DS-TWR timing.
29     rangeConfig.panId                = 0xDECA;
30     rangeConfig.initiatorAddress     = 0x0001;
31     rangeConfig.responderAddress     = 0x0002;
32     rangeConfig.responseRxAfterTxDelayUus = 1500;
33     rangeConfig.responseTxDelayUus   = 3000;
34     rangeConfig.finalTxDelayUus      = 1800;
35     rangeConfig.finalRxAfterResponseTxDelayUus = 500;
36     rangeConfig.resultRxAfterFinalTxDelayUus = 500;
37     rangeConfig.rxTimeoutUus        = 3000;
38     rangeConfig.hostTimeoutMs       = 100;
39     rangeConfig.resultRepeatCount    = 3;
40     rangeConfig.resultRepeatGapMs    = 3;
41
42     if (!uwb.begin(config, phy)) {
43         Serial.printf("UWB_BEGIN,result=FAIL,error=%s\n", uwb.lastErrorName());
44         Serial.printf("UWB_RAW_ID,dev_id=0x%08lX\n", static_cast<unsigned long>(uwb.readRawDeviceId()));
45         return false;
46     }
47
48     const uint32_t devId = uwb.deviceId();
49     Serial.printf("UWB_ID,dev_id=0x%08lX,chip=%s\n", static_cast<unsigned long>(devId), uwb.chipName());
50     if (devId != M5STAMP_UWB_QM33120_DEVICE_ID) {
51         Serial.printf("UWB_ID,result=FAIL,expected=0xDECA0314\n");
52         return false;
53     }
54
55     Serial.printf("UWB_CONFIG,result=OK,ch=%u,plen=%u,rate=6M8,tx_power=0x%08lX\n", static_cast<unsigned

```

```

56         static_cast<unsigned>(phy.preambleLength), static_cast<unsigned long>(phy.txPower));
57     return true;
58 }
59
60 static void runResponder()
61 {
62     // Wait for a Poll/Final exchange and return the DS-TWR distance result.
63     const M5Stamp_UWBDSResponderResult result = uwb.respondDSRange(rangeConfig);
64     if (!result.success) {
65         // Idle receive timeouts are expected and are not counted as failures.
66         if (result.error == M5Stamp_UWBError::RxTimeout) return;
67         if (++failCount % LOG_INTERVAL == 0) {
68             Serial.printf("DS_RESP_STAT,count=%lu,fail=%lu,last=FAIL,error=%s\n",
69                 static_cast<unsigned long>(responseCount), static_cast<unsigned long>(failCount),
70                 uwb.lastErrorName());
71         }
72         return;
73     }
74
75     // Print accumulated statistics every LOG_INTERVAL responses.
76     if (++responseCount % LOG_INTERVAL == 0) {
77         Serial.printf(
78             "DS_RESP_STAT,count=%lu,fail=%lu,last=OK,seq=%u,requester=0x%X,distance_mm=%ld,distance_m=%.
79             static_cast<unsigned long>(responseCount), static_cast<unsigned long>(failCount), result.seq
80             result.requester, static_cast<long>(result.distanceMm), result.distanceM,
81             static_cast<unsigned long>(result.elapsedMs));
82     }
83 }
84
85 void setup()
86 {
87     Serial.begin(115200);
88     delay(1000);
89     Serial.printf("M5Stamp UWB DS-TWR ANCHOR\n");
90     Serial.printf("ROLE,mode=ANCHOR\n");
91     Serial.printf("TWR_MODE,mode=DS-TWR\n");
92     uwbReady = initUwb();
93     Serial.printf("TEST_START,result=%s\n", uwbReady ? "OK" : "FAIL");
94 }
95
96 void loop()
97 {
98     if (uwbReady) runResponder();
99     else delay(1000);
100 }

```

Serial output example:

```
M5Stamp UWB DS-TWR ANCHOR
ROLE,mode=ANCHOR
TWR_MODE,mode=DS-TWR
UWB_ID,dev_id=0xDECA0314,chip=QM33120/DW3720
UWB_CONFIG,result=OK,ch=5,plen=128,rate=6M8,tx_power=0xFEFEFEFE
TEST_START,result=OK
DS_RESP_STAT,count=9500,fail=0,last=OK,seq=7,requester=0x1,distance_mm=162,distance_m=0.162,elapsed_ms=98
DS_RESP_STAT,count=9520,fail=0,last=OK,seq=27,requester=0x1,distance_mm=185,distance_m=0.185,elapsed_ms=93
DS_RESP_STAT,count=9540,fail=0,last=OK,seq=47,requester=0x1,distance_mm=181,distance_m=0.181,elapsed_ms=98
DS_RESP_STAT,count=9560,fail=0,last=OK,seq=67,requester=0x1,distance_mm=167,distance_m=0.167,elapsed_ms=98
DS_RESP_STAT,count=9580,fail=0,last=OK,seq=87,requester=0x1,distance_mm=193,distance_m=0.193,elapsed_ms=98
DS_RESP_STAT,count=9600,fail=0,last=OK,seq=107,requester=0x1,distance_mm=166,distance_m=0.166,elapsed_ms=98
DS_RESP_STAT,count=9620,fail=0,last=OK,seq=127,requester=0x1,distance_mm=164,distance_m=0.164,elapsed_ms=98
```

4. Single-Sided Two-Way Ranging (SS-TWR)

Note

This ranging method calculates the final distance using only the timestamps from the Tag, so its accuracy is strongly affected by the Tag's clock and the resulting error is relatively large. This method is not recommended. For more accurate ranging results, use Double-Sided Two-Way Ranging (DS-TWR).

4.1 Tag

cpp

```

1  #include <Arduino.h>
2  #include <M5Stamp_UWB.h>
3
4  static constexpr uint32_t RANGE_INTERVAL_MS = 200;
5  static constexpr uint32_t LOG_INTERVAL      = 10;
6
7  M5Stamp_UWB uwb;
8  M5Stamp_UWBRangeConfig rangeConfig;
9  bool uwbReady          = false;
10 uint32_t lastRangeMs = 0;
11 uint32_t rangeCount  = 0;
12 uint32_t okCount     = 0;
13
14 static bool initUwb()
15 {
16     // Stamp-C5 connections to the onboard QM33120 UWB transceiver.
17     M5Stamp_UWBConfig config;
18     config.pin_gp7      = 23;
19     config.pin_irq      = 0;
20     config.pin_wakeup   = 24;
21     config.pin_rst      = 25;
22     config.pin_miso     = 26;
23     config.pin_mosi     = 27;
24     config.pin_cs       = 11;
25     config.pin_sck      = 12;
26
27     M5Stamp_UWBPHYConfig phy;
28     phy.channel = M5Stamp_UWBChannel::Channel5;
29
30     // Both devices must use the same PAN ID, addresses, and response timing.
31     rangeConfig.panId          = 0xDECA;
32     rangeConfig.initiatorAddress = 0x0001;
33     rangeConfig.responderAddress = 0x0002;
34     rangeConfig.responseRxAfterTxDelayUus = 500;
35     rangeConfig.responseTxDelayUus      = 3000;
36     rangeConfig.rxTimeoutUus            = 6000;
37     rangeConfig.hostTimeoutMs           = 100;
38
39     if (!uwb.begin(config, phy)) {
40         Serial.printf("UWB_BEGIN,result=FAIL,error=%s\n", uwb.lastErrorName());
41         Serial.printf("UWB_RAW_ID,dev_id=0x%08lX\n", static_cast<unsigned long>(uwb.readRawDeviceId()));
42         return false;
43     }
44
45     const uint32_t devId = uwb.deviceId();
46     Serial.printf("UWB_ID,dev_id=0x%08lX,chip=%s\n", static_cast<unsigned long>(devId), uwb.chipName());
47     if (devId != M5STAMP_UWB_QM33120_DEVICE_ID) {
48         Serial.printf("UWB_ID,result=FAIL,expected=0xDECA0314\n");
49         return false;
50     }
51
52     Serial.printf("UWB_CONFIG,result=OK,ch=%u,plen=%u,rate=6M8,tx_power=0x%08lX\n", static_cast<unsigned long>(uwb.chipName()), static_cast<unsigned long>(phy.preambleLength), static_cast<unsigned long>(phy.txPower));
53     return true;
54 }
55

```

```

56
57 static void runRanging()
58 {
59     // Start one SS-TWR exchange at the configured interval.
60     if (millis() - lastRangeMs < RANGE_INTERVAL_MS) return;
61     lastRangeMs = millis();
62
63     const M5Stamp_UWBRangeResult result = uwb.requestRange(rangeConfig);
64     rangeCount++;
65     okCount += result.success;
66
67     // Print accumulated statistics every LOG_INTERVAL attempts.
68     if (rangeCount % LOG_INTERVAL) return;
69
70     const uint32_t failCount = rangeCount - okCount;
71     if (result.success) {
72         Serial.printf(
73             "SS_RANGE_STAT,count=%lu,ok=%lu,fail=%lu,last=OK,seq=%u,distance_mm=%ld,distance_m=%.3f,elap
74             static_cast<unsigned long>(rangeCount), static_cast<unsigned long>(okCount),
75             static_cast<unsigned long>(failCount), result.sequence, static_cast<long>(result.distanceMm)
76             result.distanceM, static_cast<unsigned long>(result.elapsedMs));
77     } else {
78         Serial.printf("SS_RANGE_STAT,count=%lu,ok=%lu,fail=%lu,last=FAIL,seq=%u,error=%s\n",
79             static_cast<unsigned long>(rangeCount), static_cast<unsigned long>(okCount),
80             static_cast<unsigned long>(failCount), result.sequence, uwb.lastErrorName());
81     }
82 }
83
84 void setup()
85 {
86     Serial.begin(115200);
87     delay(1000);
88     Serial.printf("M5Stamp UWB SS-TWR TAG\n");
89     Serial.printf("ROLE,mode=TAG\n");
90     Serial.printf("TWR_MODE,mode=SS-TWR\n");
91     uwbReady = initUwb();
92     Serial.printf("TEST_START,result=%s\n", uwbReady ? "OK" : "FAIL");
93 }
94
95 void loop()
96 {
97     if (uwbReady) runRanging();
98     else delay(1000);
99 }

```

Serial output example:

```
M5Stamp UWB SS-TWR TAG
ROLE,mode=TAG
TWR_MODE,mode=SS-TWR
UWB_ID,dev_id=0xDECA0314,chip=QM33120/DW3720
UWB_CONFIG,result=OK,ch=5,plen=128,rate=6M8,tx_power=0xFEFEFEFE
TEST_START,result=OK
SS_RANGE_STAT,count=10,ok=10,fail=0,last=OK,seq=10,distance_mm=2270,distance_m=2.270,elapsed_ms=4
SS_RANGE_STAT,count=20,ok=20,fail=0,last=OK,seq=20,distance_mm=2282,distance_m=2.282,elapsed_ms=4
SS_RANGE_STAT,count=30,ok=30,fail=0,last=OK,seq=30,distance_mm=2275,distance_m=2.275,elapsed_ms=4
SS_RANGE_STAT,count=40,ok=40,fail=0,last=OK,seq=40,distance_mm=2251,distance_m=2.251,elapsed_ms=4
SS_RANGE_STAT,count=50,ok=50,fail=0,last=OK,seq=50,distance_mm=2265,distance_m=2.265,elapsed_ms=4
SS_RANGE_STAT,count=60,ok=60,fail=0,last=OK,seq=60,distance_mm=2301,distance_m=2.301,elapsed_ms=4
SS_RANGE_STAT,count=70,ok=70,fail=0,last=OK,seq=70,distance_mm=2301,distance_m=2.301,elapsed_ms=4
SS_RANGE_STAT,count=80,ok=80,fail=0,last=OK,seq=80,distance_mm=2291,distance_m=2.291,elapsed_ms=4
SS_RANGE_STAT,count=90,ok=90,fail=0,last=OK,seq=90,distance_mm=2282,distance_m=2.282,elapsed_ms=4
SS_RANGE_STAT,count=100,ok=100,fail=0,last=OK,seq=100,distance_mm=2312,distance_m=2.312,elapsed_ms=4
```

4.2 Anchor

cpp

```

1  #include <Arduino.h>
2  #include <M5Stamp_UWB.h>
3
4  static constexpr uint32_t LOG_INTERVAL = 20;
5
6  M5Stamp_UWB uwb;
7  M5Stamp_UWBRangeConfig rangeConfig;
8  bool uwbReady          = false;
9  uint32_t responseCount = 0;
10 uint32_t failCount      = 0;
11
12 static bool initUwb()
13 {
14     // Stamp-C5 connections to the onboard QM33120 UWB transceiver.
15     M5Stamp_UWBConfig config;
16     config.pin_gp7      = 23;
17     config.pin_irq      = 0;
18     config.pin_wakeup   = 24;
19     config.pin_rst      = 25;
20     config.pin_miso     = 26;
21     config.pin_mosi     = 27;
22     config.pin_cs       = 11;
23     config.pin_sck      = 12;
24
25     M5Stamp_UWBPHYConfig phy;
26     phy.channel = M5Stamp_UWBChannel::Channel5;
27
28     // Both devices must use the same PAN ID, addresses, and response timing.
29     rangeConfig.panId          = 0xDECA;
30     rangeConfig.initiatorAddress = 0x0001;
31     rangeConfig.responderAddress = 0x0002;
32     rangeConfig.responseRxAfterTxDelayUus = 500;
33     rangeConfig.responseTxDelayUus      = 3000;
34     rangeConfig.rxTimeoutUus            = 6000;
35     rangeConfig.hostTimeoutMs           = 100;
36
37     if (!uwb.begin(config, phy)) {
38         Serial.printf("UWB_BEGIN,result=FAIL,error=%s\n", uwb.lastErrorName());
39         Serial.printf("UWB_RAW_ID,dev_id=0x%08lX\n", static_cast<unsigned long>(uwb.readRawDeviceId()));
40         return false;
41     }
42
43     const uint32_t devId = uwb.deviceId();
44     Serial.printf("UWB_ID,dev_id=0x%08lX,chip=%s\n", static_cast<unsigned long>(devId), uwb.chipName());
45     if (devId != M5STAMP_UWB_QM33120_DEVICE_ID) {
46         Serial.printf("UWB_ID,result=FAIL,expected=0xDECA0314\n");
47         return false;
48     }
49
50     Serial.printf("UWB_CONFIG,result=OK,ch=%u,plen=%u,rate=6M8,tx_power=0x%08lX\n", static_cast<unsigned>(phy.preambleLength), static_cast<unsigned long>(phy.txPower));
51     return true;
52 }
53
54
55 static void runResponder()

```



```

56 {
57     // Wait for a Poll frame and send the delayed SS-TWR Response frame.
58     const M5Stamp_UWBResponderResult result = uwb.respondRange(rangeConfig);
59     if (!result.success) {
60         // Idle receive timeouts are expected and are not counted as failures.
61         if (result.error == M5Stamp_UWBError::RxTimeout) return;
62         if (++failCount % LOG_INTERVAL == 0) {
63             Serial.printf("SS_RESP_STAT,count=%lu,fail=%lu,last=FAIL,error=%s\n",
64                 static_cast<unsigned long>(responseCount), static_cast<unsigned long>(failCount),
65                 uwb.lastErrorName());
66         }
67         return;
68     }
69
70     // Print accumulated statistics every LOG_INTERVAL responses.
71     if (++responseCount % LOG_INTERVAL == 0) {
72         Serial.printf("SS_RESP_STAT,count=%lu,fail=%lu,last=OK,seq=%u,requester=0x%X,elapsed_ms=%lu\n",
73             static_cast<unsigned long>(responseCount), static_cast<unsigned long>(failCount),
74             result.requester, static_cast<unsigned long>(result.elapsedMs));
75     }
76 }
77
78 void setup()
79 {
80     Serial.begin(115200);
81     delay(1000);
82     Serial.printf("M5Stamp UWB SS-TWR ANCHOR\n");
83     Serial.printf("ROLE,mode=ANCHOR\n");
84     Serial.printf("TWR_MODE,mode=SS-TWR\n");
85     uwbReady = initUwb();
86     Serial.printf("TEST_START,result=%s\n", uwbReady ? "OK" : "FAIL");
87 }
88
89 void loop()
90 {
91     if (uwbReady) runResponder();
92     else delay(1000);
93 }

```

Serial output example:

```

M5Stamp UWB SS-TWR ANCHOR
ROLE,mode=ANCHOR
TWR_MODE,mode=SS-TWR
UWB_ID,dev_id=0xDECA0314,chip=QM33120/DW3720
UWB_CONFIG,result=OK,ch=5,plen=128,rate=6M8,tx_power=0xFEFEFEFE
TEST_START,result=OK
SS_RESP_STAT,count=1260,fail=0,last=OK,seq=12,requester=0x1,elapsed_ms=98
SS_RESP_STAT,count=1280,fail=0,last=OK,seq=32,requester=0x1,elapsed_ms=98
SS_RESP_STAT,count=1300,fail=0,last=OK,seq=52,requester=0x1,elapsed_ms=93
SS_RESP_STAT,count=1320,fail=0,last=OK,seq=72,requester=0x1,elapsed_ms=98
SS_RESP_STAT,count=1340,fail=0,last=OK,seq=92,requester=0x1,elapsed_ms=98

```

5. Compile and Upload

- Select the device port, paste one of the example sketches above into the Arduino IDE, click the compile/upload button in the upper-left corner of the Arduino IDE, and wait for the sketch to finish compiling and upload to the device.

