

# When EMP Throws SCCIC Timeout

When we try to read a register in the IpGBT we get the following error,

```
emp._emp_python.exception: SCCICNode: Timed out waiting for replies
(expecting 1 words, received 0 so far)
```

This report will outline when the emp C++ layer in the MTD DAQ SW throws this error.

This error is thrown in the `SCCICNode.cpp` (in path `/home/cmx/etl/toolbox/mtd-emp-toolbox/core/src/common/SCCICNode.cpp`) file in the method `waitReplyDataCount`,

```
bool SCCICNode::waitReplyDataCount(unsigned aNData, int aTmax_msec) const
{
    unsigned lI(0);
    char lBuf[256];
    sprintf(lBuf, "SCCICNode::waitReplyDataCount() Expected
reply_data_count=%d", aNData);
    logger::debug(lBuf);

    using namespace std::chrono;
    const auto lStart = steady_clock::now();
    while (true) {

        auto start = std::chrono::steady_clock::now();
        ValWord<uint32_t> lNReply = getNode("reply_data_count").read();
        getClient().dispatch();
        auto end = std::chrono::steady_clock::now();
        SCCICNode::m_read_reply_data_count_dt += (end - start);
        sprintf(lBuf, "SCCICNode::waitReplyDataCount() reply_data_count=%d",
lNReply.value());
        logger::debug(lBuf);

        if (lNReply < aNData) {
            if (!aTmax_msec != -1 && lI == kTmax * aNData) {
                readReply(lNReply);
                ICTimeOut lExc("SCCICNode: Timed out waiting for replies (expecting
" + std::to_string(aNData) + " words, received " +
std::to_string(lNReply.value()) + " so far)");
                throw lExc;
            }
            std::this_thread::sleep_for(std::chrono::milliseconds(1));
        }
    }
}
```

```

    }
    else if (lNReply == aNData)
        return true;
    else {
        readReply(lNReply);
        IInvalidNumberOfReplies lExc("Reply_data_count is not valid.");
        throw lExc;
    }
    lI++;
    if (aTmax_msec != -1 && duration_cast<milliseconds>(steady_clock::now()
- lStart).count() >= aTmax_msec)
        return false;
    }
}

```

For thoroughness this is called at the end of `icReadBlock` and in `icWriteBlock`. These methods are ultimately used in the top level python software. By studying this we hope to understand what counts as a timeout. Specifically this line,

```

ICTimeout lExc("SCCICNode: Timed out waiting for replies (expecting " +
std::to_string(aNData) + " words, received " +
std::to_string(lNReply.value()) + " so far)");

```

## How it works

The function expects some data length for the reply and begins reading reply data counts,

```

ValWord<uint32_t> lNReply = getNode("reply_data_count").read();

```

If length of the reply is less than the inputted expected it throws this error. However it will try `10 * len(expected_data)` iterations before timing out.

```

if (lNReply < aNData) {
    if (!aTmax_msec != -1 && lI == kTmax * aNData) {
        readReply(lNReply);
        ITimeout lExc("SCCICNode: Timed out waiting for replies (expecting " +
std::to_string(aNData) + " words, received " +
std::to_string(lNReply.value()) + " so far)");
        throw lExc;
    }
}

```

```
std::this_thread::sleep_for(std::chrono::milliseconds(1));  
}
```