

Testování PS6000 a GUI oscilloscope device

Datum: 20.5.2024

Commit: 5f44a490d892934aa1b5b2777ca48ec2abd87001

Prostředí: Windows 10 Pro, Qt Creator 10.0.1, Qt 6.5.0 MinGW 64-bit

Shrnutí:

Při práci s jedním kanálem: program funguje dobře.

Při práci s dvěma kanály: program padá extrémně často a není zřejmé proč a kde. Debugger ukazuje, že se tak děje ve funkci `TScopeCollector::collectData` na řádku `buffer = new uint8_t [bufferSize];`. Chtělo by to vyzkoušet program s memory debuggerem. Program spadne téměř vždy při zmáčknutí tlačítka stop (ale když trigger není nastaven na kanál A). Dále pak při měření více záznamů.

Nemohli jsme dostatečně všechno otestovat, protože program padá příliš často.

PS6000 + GUI

Všechny následující body platí pro aktivní trigger a dva enabled kanály.

- Plugin indexuje kanály od 1, ale GUI indexuje od 0. Pro první channel se zavolá `downloadSamples` s `channel=0`, což v pluginu ve switch větvení vyvolá `default expression` a vrátí správně první kanál. Pro druhý kanál (tedy `channel=1`) se rovněž vrátí první kanál.
 - Patrně místo `downloadSamples (i, ...` máš volat
`downloadSamples (status [i].m_index, ...`
- Pokud kliknu na stop (ať už během jednoho měření, tak během opakovaného měření), tak program spadne na následujícím místě:

```

    size_t groupSampleSize = sampleCount / idealPointCount;
    idealPointCount: 691
    great min, max;
    size_t sampleIndex = sampleOffset;
    while(sampleIndex < sampleOffset + sampleCount) {
        sampleOffset += 5
        min = firstValue;
        max = firstValue;
        max += slope;
        min -= slope;
        size_t loopEndIndex = (sampleIndex + groupSampleSize) > sampleCount ? sampleCount : sampleIndex + groupSampleSize;
        for (size_t i = sampleIndex + 1; i < loopEndIndex; ++i) {
            sampleIndex = 5
            sampleOffset = 99449320
            min = value - (buffer[i] - channel.getMinValue()) - channel.getRange() + channel.getOffset();
            max = value + (buffer[i] - channel.getMinValue()) - channel.getRange() + channel.getOffset();
            pointList.append(QPointF(sampleIndex, min));
            pointList.append(QPointF(sampleIndex, max));
        }
    }
    sampleIndex == groupSampleSize
}

```

- Chybná data má program už v TSscopeWidget::receiveTraces, protože debugger ukazuje typ TUint8, které je špatně pro picoscope.
- Pokud je trigger na kanál A, pak program nepadá. Pro ostatní kanály ano.
- Při měření více záznamů program spadne
 - Nastavím si měření (jeden záznam je dlouhý 15626) a nastavím počet záznamu na 20. Dám měřit jednou (to funguje). Pak dám měřit znova a program spadne. Debugger ukazuje dvě místa pádu (pouštěl jsme opakovaně a někdy se vyhodí jedno místo a někdy druhé):

```

    uint32_t capturesR;
    status = ps6000GetNoOfCaptures(m_handle, &capturesR);
    if (status || capturesR != m_captures) {
        qWarning("Failed to capture specified number of traces per run");
        return 0;
    }

    for (uint32_t i = 0; i < m_captures; ++i) {
        status = ps6000SetDataBufferBulk(m_handle, psChannel, reinterpret_cast<short*>(buffer) + i * psSamples, psSamples);
        if (status) {
            qWarning("Failed to set up receiving buffer");
            return 0;
        }

        if (psSamples != (m_preTrigSamples + m_postTrigSamples)) {
            qWarning("Failed to receive the data");
            return 0;
        }
    }

    if (status = ps6000GetValuesBulk(m_handle, &psSamples, 0, m_captures - 1, PS6000_RATIO_MODE_NONE, over.get());)
        if (status || psSamples != (m_preTrigSamples + m_postTrigSamples)) {
            qWarning("Failed to receive the data");
            return 0;
        }

    status = ps6000GetValues(m_handle, 0, &psSamples, 1, PS6000_RATIO_MODE_NONE, 0, over.get());
    if (status || psSamples != (m_preTrigSamples + m_postTrigSamples)) {
        qWarning("Failed to receive the data");
        return 0;
    }
}

```

The screenshot shows the Qt Creator IDE interface during debugging. The code editor displays `tscopemodel.cpp` with a break point at line 189. The registers and memory dump panes show various variables and their values. The assembly and registers pane at the bottom shows the instruction flow and register states.

- Pád programu u funkce new naznačuje špatnou práci s pamětí (někde jinde se přepisují data nebo něco podobného). Tento pád jsem pozoroval i u jedné tracy (ale i u více), ale je to spíš na náhodě, kdy se tak stane. Chtělo by to otestovat pomocí memory debuggeru (na windows ho nemám).
- Pokud je trigger disabled, stále se ukazuje šípička v grafu.