

Article

Automated DNA Extraction Monitoring System Based on MTConnect Technology

Sang-Ho Han ¹, Ae-Ja Park ², Ah-Reum Park ² and Mun-Ho Ryu ^{3,4,*} ¹ Department of Healthcare Engineering, Jeonbuk National University, Jeonju 54896, Korea; enki@jbnu.ac.kr² Department of Bioinstrument, Biomedux Co., Suwon 16226, Korea; ajcp@biomedux.com (A.-J.P.); arpark@biomedux.com (A.-R.P.)³ Division of Biomedical Engineering, Jeonbuk National University, Jeonju 54896, Korea⁴ Research Center of Healthcare & Welfare Instrument for the Aged, Jeonbuk National University, Jeonju 54896, Korea

* Correspondence: mhryu@jbnu.ac.kr

Abstract: MTConnect standard technology provides simplicity, flexibility, and scalability in integrating various equipment and operating systems and enabling accurate and consistent data collection from any MTConnect-compatible system. Using MTConnect technology, it is possible to immediately identify the cause of a problem and respond quickly when a problem occurs. Molecular genetic diagnostic point-of-care testing (POCT) devices have received attention in recent years because they enable rapid disease diagnosis. A molecular genetic diagnostic POCT device is under development by the authors. The system consists of a gene extraction process and a real-time PCR-based gene amplification process. In this study, we propose and demonstrate a system based on MTConnect technology to monitor an automated DNA extraction process. The proposed system consists of an automated DNA extraction system, an MTConnect adapter, an MTConnect agent, and a client application. The adapter and agent were developed on a Raspberry Pi single-board computer. The agent publishes the collected data in Extensible Markup Language (XML) format over a network. The performance and reliability of the system were evaluated by verifying the request response time between the implemented system's agent and the client application. The results demonstrate the feasibility of monitoring the DNA extraction process over a network.



Citation: Han, S.-H.; Park, A.-J.; Park, A.-R.; Ryu, M.-H. Automated DNA Extraction Monitoring System Based on MTConnect Technology. *Appl. Sci.* **2021**, *11*, 684. <https://doi.org/10.3390/app11020684>

Received: 6 December 2020

Accepted: 10 January 2021

Published: 12 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: molecular genetic test; DNA extraction; POCT device; MTConnect; real-time monitoring

1. Introduction

At the core of the smart factory, according to the Industry 4.0 trend, is a technology that enables connection between devices [1,2]. MTConnect standard technology provides simplicity, flexibility, and scalability in integrating various equipment and operating systems and enabling accurate and consistent data collection from any MTConnect-compatible system. It can be used to collect and analyze a large amount of equipment data and monitor them over a network. Using MTConnect technology, it is possible to immediately identify the cause of a problem and respond quickly when a problem occurs. For this purpose, it is necessary to monitor the status of equipment and send status information to the manager in the event of a problem.

Since 2007, with the support of the Association for Manufacturing Technology, U.C. Berkeley has been developing a standard protocol and integration method for information exchange between various pieces of production equipment, and as a result, MTConnect was proposed. With this revision and supplementation, MTConnect was established as an open standard based on Transmission Control Protocol/ Internet Protocol (TCP/IP) by the MTConnect Institute [3]. This makes it possible to access the status information and data on different facilities produced by various manufacturers in a standardized form outside the device. The MTConnect standard enables the implementation of a device-independent and highly scalable monitoring system because detailed information on the equipment

requiring the monitoring system can be obtained in Extensible Markup Language (XML) format. As is well known, robots are embedded with sensors to allow for machine–robot or human–robot collaboration in a safe workspace of smart factories. MTConnect standard technology is applicable in the integration of components of smart factories, such as machines and robots [4–9]. As a result, more than 500 organizations, including major US and Japanese companies, currently support the standard.

In recent years, the healthcare market has demanded rapid diagnosis and treatment according to diagnosis results, and the types of diagnostic devices suitable for meeting these needs are increasing. Point-of-care testing (POCT) devices have attracted attention in recent years because they enable rapid disease diagnosis. In the field of on-site diagnosis, the trend is to expand to other inspection items by developing technology that prevents errors even when generalists rather than specialists are involved. This also applies to situations that require quick inspection results.

World-class manufacturers in the field of diagnostic laboratory testing equipment have released many field-testing products to meet these needs. Some manufacturers and startups are investing more in POCT field inspection equipment development than in traditional inspection fields. The next-generation molecular diagnosis field is seeing improvements in the accuracy and efficiency of diagnosis based on digital PCR technology. In the field of field diagnosis equipment, microelectromechanical systems (MEMS) and nanotechnology have been applied to miniaturization, and efforts are being made to effectively apply these technologies to the diagnosis and prevention of diseases through the control, inspection, and data management of diagnosis equipment through convergence with IT technology. This trend may be a result of the various advantages of POCT (e.g., shortening of examination times, patient satisfaction, shortened hospitalization times, improved visibility, reduced expenses, and improved preparedness for emergencies) [10–13].

POCT equipment has been linked to the telemedicine system, which allows patients to communicate with doctors from their homes. This is a major transformation of the clinical pathological examination system. Confidence in the reliability of POCT equipment is crucial to the equipment's effectiveness in telemedicine systems. POCT devices must be remotely controlled by a connection to the examination information systems of the medical institution, and the results must be collected, reviewed for quality control purposes, and automatically entered into the patient's electronic medical record. Therefore, it is very important to ensure the reliability of diagnosis results for a medical institution to be able to remotely monitor the state of POCT equipment at a remote location in real time. There have been few studies on the remote monitoring of medical devices, including POCT equipment [14,15]. However, these medical device monitoring systems simply monitor the measurement results or simple operation status. Therefore, this study proposes a system that can monitor detailed elements of device components with MTConnect standard technology.

In this paper, an MTConnect technology-based monitoring system for automated molecular genetic diagnostic test devices is presented. This automated DNA extraction system consists of a component for gene extraction and gene amplification based on real-time PCR and a Raspberry Pi-based control system that controls a series of extraction procedures. The user can control the system using a web application-based user control interface. In addition, MTConnect, which is an open and standard machine monitoring data interface based on XML [16], was used to enable POCT device monitoring data collection and monitoring of POCT devices as web application-based clients.

This paper is organized into five sections. The method for system design, automated DNA extraction system, and MTConnect is in Section 2. The experimental results are in Section 3, and discussions are reported in Section 4, and finally, the conclusion is presented in Section 5.

2. Materials and Methods

The proposed system consists of an automated DNA extraction system, an MTConnect adapter based on the MTConnect standard technology, an MTConnect agent, and a user-

controlled UI client application based on a web application. Figure 1 shows the working principle of MTConnect with the POCT device.

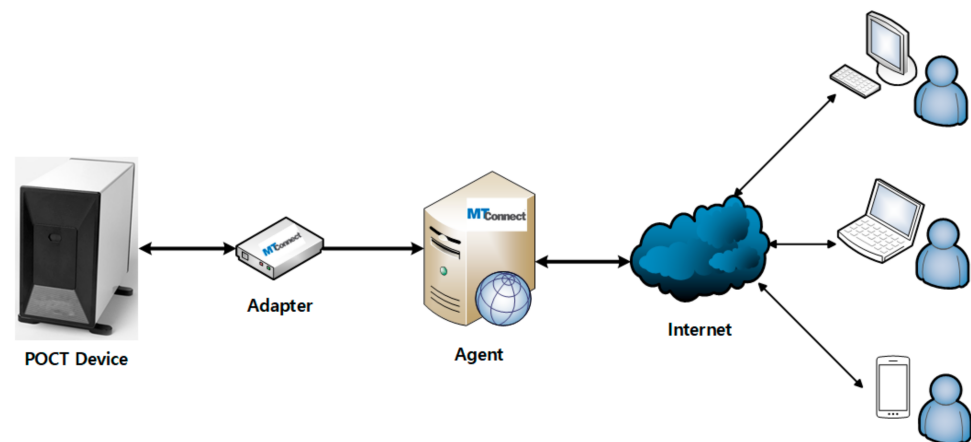


Figure 1. Working principle of MTConnect with point-of-care testing (POCT) device.

2.1. Automated DNA Extraction System

A molecular genetic diagnostic POCT device is under development by the authors. The system consists of a gene extraction process and a real-time PCR-based gene amplification process. An automated DNA extraction system configuration diagram is shown in Figure 2.

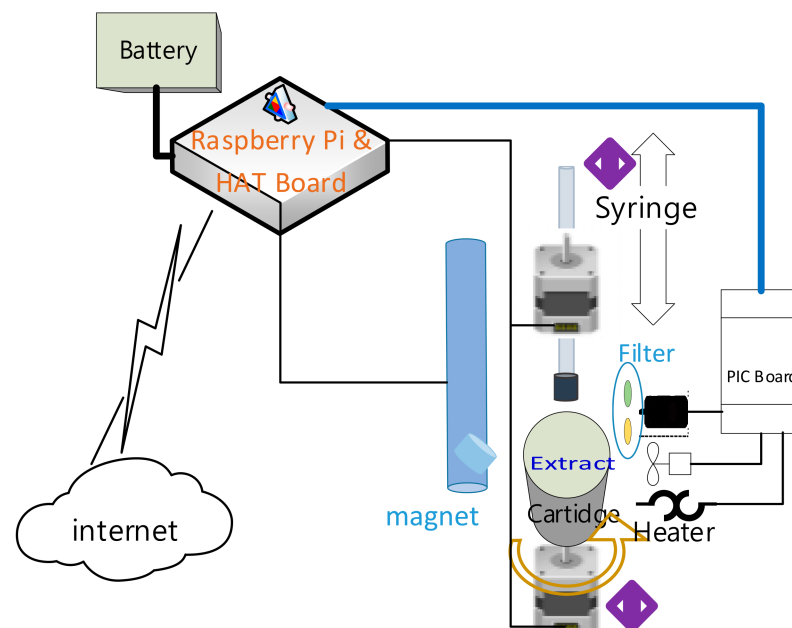


Figure 2. DNA extraction system configuration diagram. The PIC microcontroller (PIC) Board is connected to the Hardware Attached on Top (HAT) Board.

The DNA extraction mechanism consists of a cartridge with chambers, syringes, four linear stepper actuators, and a rotary stepper actuator. The actuators provide a sequence of steps in the DNA extraction process, including transporting, mixing, and washing to obtain a gene specimen, magnetic bead, and reagent solution. A control application programmed in Python compiles the DNA extraction sequence protocol into binary instructions, triggers the execution of instructions on a controller implemented in Raspberry Pi 3B, and displays the execution status.

The Raspberry Pi-based control system interfaces with the web application-based user control UI through TCP/IP communication and executes protocol sequence commands

triggered by the web application-based user control UI. With respect to the axis operation command, the stepper motor controller produces an add-on board in the form of a HAT board that applies the L6470 Stepper Driver IC and connects it to the Raspberry Pi 3B. The L6470 Stepper Driver IC (STMicroelectronics Inc., Geneva, Switzerland) provides smooth linear acceleration and velocity profiles for stepper motor drivers. To establish the mechanical position, the controller provides a homing function that executes an axis motion until coming into contact with a home switch and sets that position as the predefined position. The controller also provides safety checks, such as positive and negative limit checking functions, by which it prevents any axis motion beyond the positions governed by the limit switches [17].

To evaluate the miniaturization of the automated DNA extraction monitoring system and the usability for portable operation, a system power supply for a portable battery power HAT board was added, along with other component control and function expansion boards, to ensure flexibility in the system hardware configuration.

The system was implemented as a web server with Flask, and a web application-based user control UI was constructed using Vue. JS was used for two-way JavaScript Object Notation (JSON) communication with the motor drive system control driver through ZeroMQ (ZMQ) and to run the automated DNA extraction system. HTTP communication between the web application client and the web server was implemented with Axios, an Ajax library that specializes in handling only asynchronous HTTP requests. Because the web application-based user control UI configured as described above is equipped with a fully automatic system function, it is possible to perform molecular genetic diagnostic tests without specialized knowledge about genetic analysis. The automated DNA extraction system block diagram is shown in Figure 3.

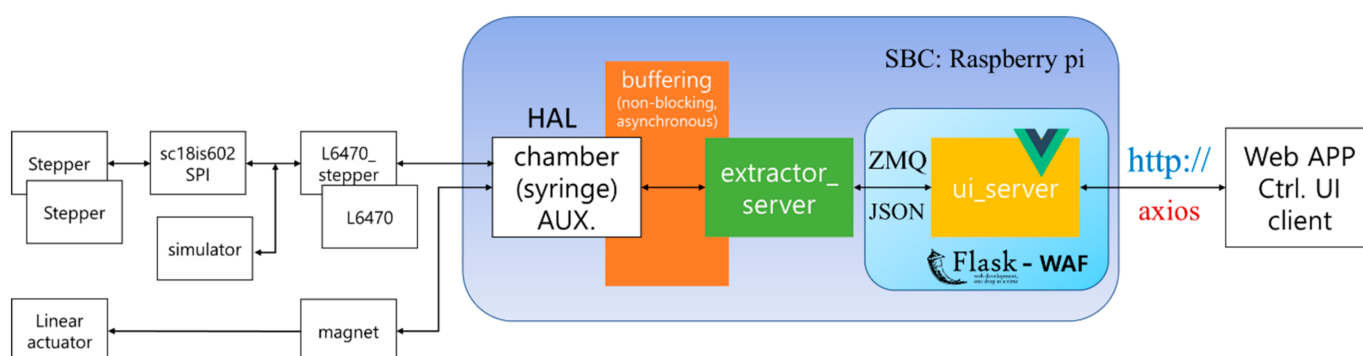


Figure 3. DNA extraction system block diagram. A hardware abstraction layer (HAL) directly drives the motor driver and auxiliary (AUX.) devices such as Linear actuator and magnet. Flask is a micro web application framework (WAF) written in Python. It communicates HTTP with the web application control (Ctrl.) UI client.

2.2. MTConnect Standard

MTConnect is a lightweight scalable standard protocol that defines the interface between devices and machine tools and information systems or applications to collect and manage manufacturing data [18]. MTConnect consists of devices, adapters, agents, networks, and client applications. Unlike Open Platform Communications (OPC), MTConnect provides easily scalable protocols and data expression and exchange methods between production equipment and various applications, but it only specifies a one-way communication protocol that flows from a device to an application.

As a software component of MTConnect, there is an agent at the center of communication, and an adapter is implemented for each of the various existing devices or sensors to relay communication between the device and the agent. The agent transmits the status of the device to the monitoring/analysis application program using a standard web protocol and XML. The data in XML delivered by MTConnect for data exchange are defined by three types of standard schemas: a device schema that describes the type and configuration of the

equipment, a stream schema that describes the state of the general data during operation of the equipment, and an error schema that provides information about errors or warnings. A system for monitoring multiple devices follows these schemas and MTConnect communication protocols. When a new device is added, the data for the new device are described only within the context of the exchanged XML document data, without overall modification of the system, enabling flexible expansion [19].

In MTConnect, the client requesting data and the server (or agent) providing data communicate through an HTTP protocol, and the data exchanged over HTTP are in XML format. The main elements of XML defined in MTConnect can be extended by users. The MTConnect agent stores data from manufacturing facilities in memory or storage by first in, first out (FIFO) means, and when a request from a client is received, the agent provides an XML document containing predefined elements as a response to the request. The elements and content contained in the XML document depend on the type of request. The types of requests that the MTConnect client can send to the MTConnect agent are probe, current, and sample requests.

2.3. MTConnect with Automated DNA Extraction System

The MTConnect adapter and agent were developed on a Raspberry Pi single-board computer. The adapter collects and sends information on the state of the extraction process in the MTConnect standard format. The agent publishes the collected data in XML format through the network and monitors them in a web application acting as an MTConnect client.

2.3.1. MTConnect Adapter

The MTConnect adapter consists of a transmission unit and a connection unit that collects status data on the target equipment between the monitored equipment and the MTConnect agent and sends status data whenever the MTConnect agent requests it.

The method is implemented in the form of an MTConnect agent in Python 3.7, with reference to the “Makerbot Agent-Adapter” source in the official github repertoire of the MTConnect Institute [20]. Upon receiving a request from the MTConnect agent, the MTConnect adapter accesses the application programming interface (API) of the control program of the Automated DNA Extraction System and obtains the current hardware status information received after transmitting the request. Specifically, it connects to a specific address of the automated DNA extraction system control program and a ZeroMQ-based socket on a specific port, sends a request to JSON according to the format specified in the predefined API, receives the resulting message, and parses it. Data are classified and saved by parameter and returned to the MTConnect agent. Figure 4 is a flow chart showing how the automated DNA extraction system processes the response of the MTConnect adapter in processing the received DNA processing command.

2.3.2. MTConnect Agent

The MTConnect agent collects data from the MTConnect adapter running on the target device (or the computer connected to the device) to be monitored. Whenever the MTConnect client receives a request from a running device, the collected data are processed and provided in XML format according to the request type.

MTConnect can express metric data for use in the operation of an on-machine configuration system. MTConnect documents consist of dynamic data in the form of time series with time stamps. The MTConnect agent generates raw data collected from machine tool controllers and measuring devices, in compliance with the MTConnect protocol. From these MTConnect documents, we collect instances of attribute data that users need. MTConnect documents can be obtained either directly from the MTConnect agent or indirectly from external databases and manufacturing equipment applications.

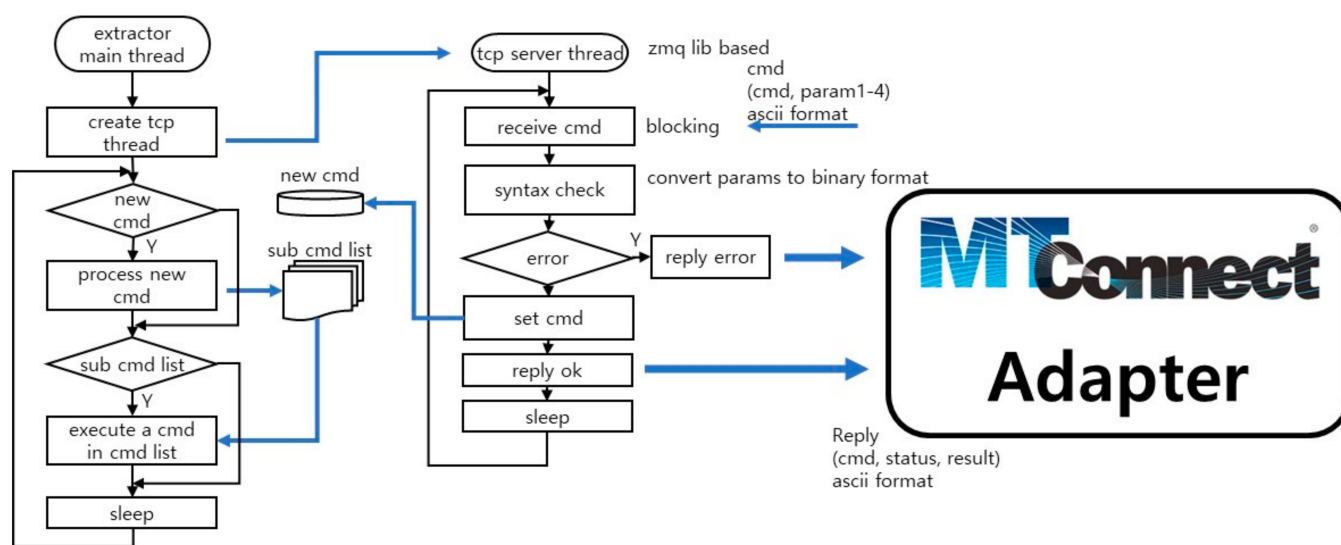


Figure 4. DNA extraction process command (cmd) with parameter (param) flow chart.

In the automated DNA extraction system's web application-based user control UI server program, a dedicated API web socket that transmits requests to and processes responses from the MTConnect agent is added and initialized, and the MTConnect agent resides in memory and starts to operate. When the web app client requests MTConnect data, the web server program sends the relevant request to the agent in JSON format, receives the response data, and then transmits the XML data according to the MTConnect standard format received by the agent to the client that requested the data.

MTConnect is a standard originally developed to interface between information systems or applications with production equipment and machine tools to collect and manage manufacturing data. Many of the general "DataItems" specified in the standard are far from those optimized for the characteristics of medical devices. Therefore, "DataItem," the specific device component of the automated DNA extraction system to be monitored through MTConnect, was selected as an element suitable for the characteristics of POCT equipment, based on the latest MTConnect Version 1.6 standard specification and implemented in the agent. Figure 5 is a schematic diagram of the information models of the automated DNA extraction system, organized according to the relationships of major components.

2.4. MTConnect Monitoring Client Web Application

The MTConnect monitoring client receives XML data from an agent that complies with the MTConnect standard, processes it into a format that is easy for users to see, and displays it on the GUI screen of the user device.

Although the MTConnect standard transmits data in XML format, which is a well-organized hierarchical structure, most of the data read from the agent by the MTConnect client is in a structure that is difficult for users to see and understand at a glance. Figure 6 is an example of an instance of the collection of attribute data related to the position and speed of a specific driving unit from the MTConnect XML document.

The data received by the MTConnect client are extracted using the XML parsing method and then visualized and displayed to the user in real time.

The MTConnect monitoring client application is implemented with elements and structures similar to the web application-based user control UI of the automated DNA extraction system. Figure 7 is a block diagram showing how the MTConnect data flow in processing the response of the web application command.

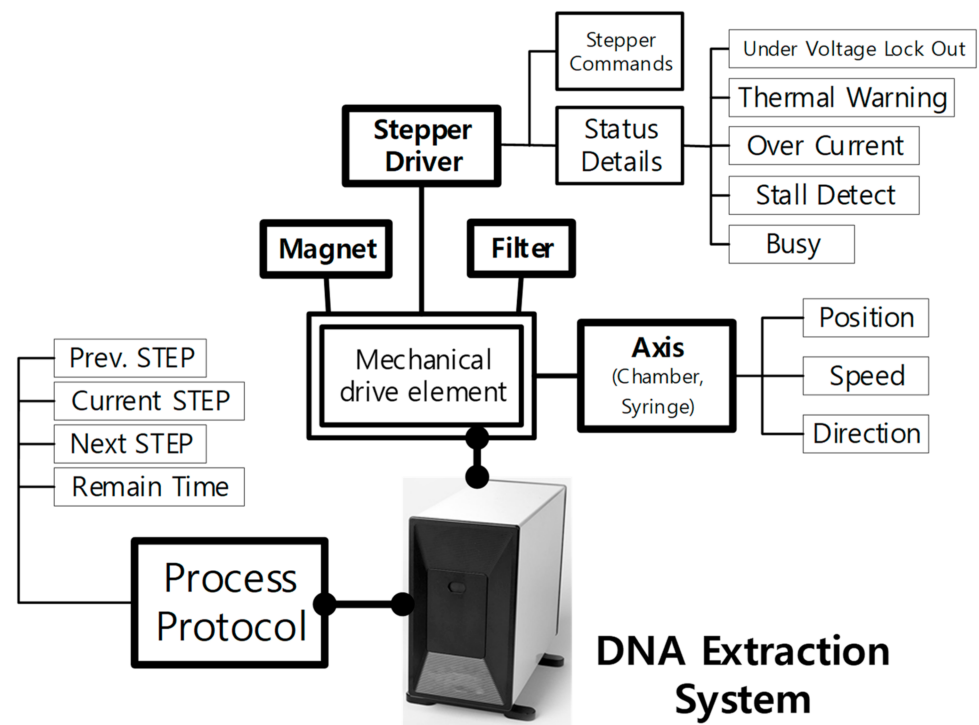


Figure 5. Information model of DNA extraction system.

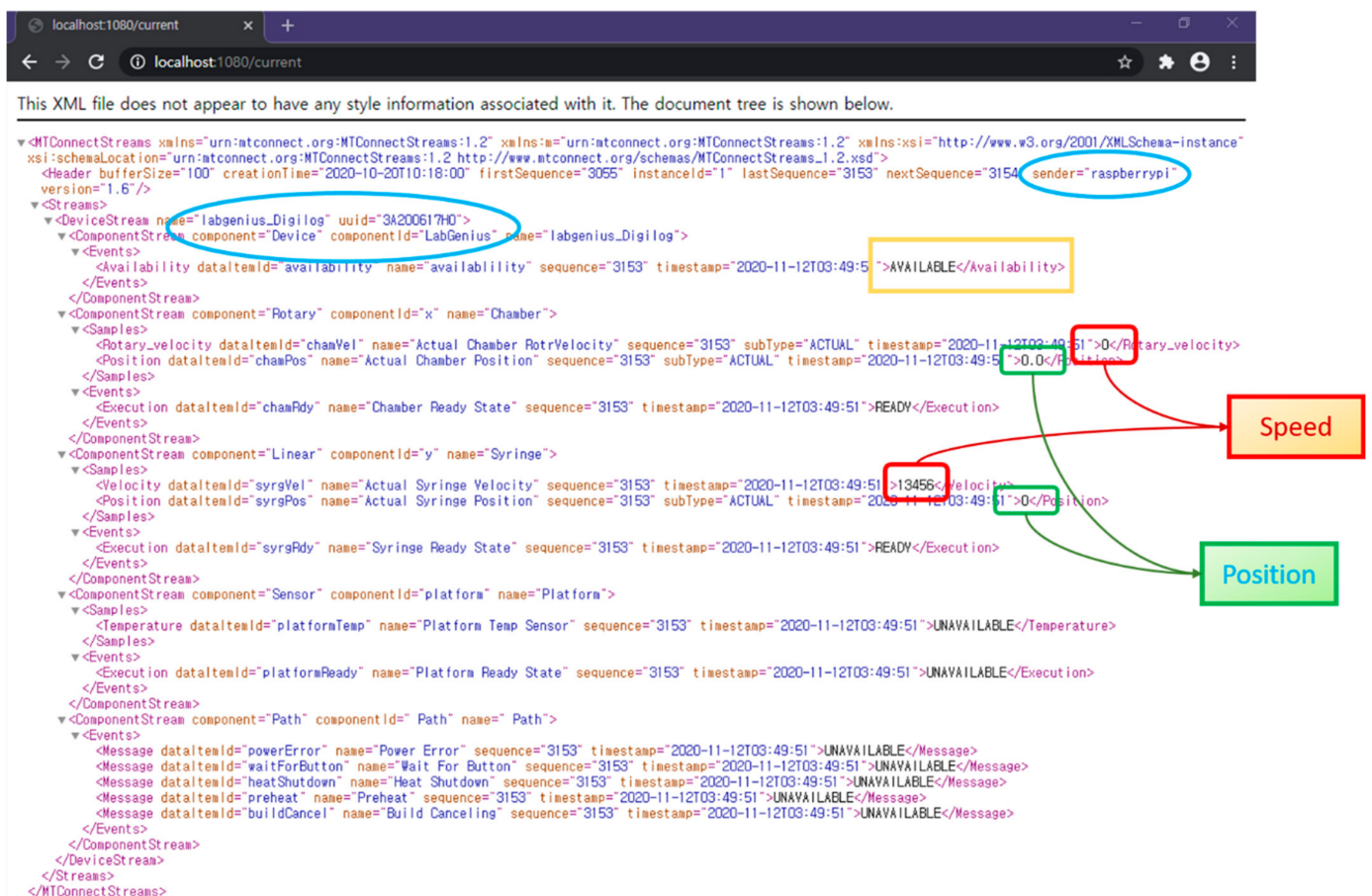


Figure 6. MTConnect Extensible Markup Language (XML) for "Current" command.

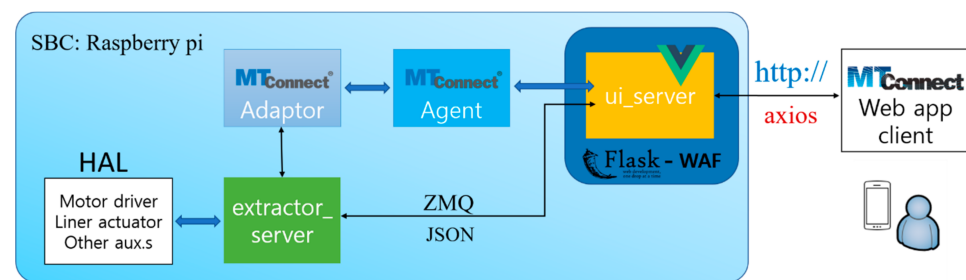


Figure 7. MTConnect data flow.

3. Results

The MTConnect monitoring client presented in this paper was verified in operation by interworking with the MTConnect agent installed on the Raspberry Pi 3B. The automated DNA extraction system measured the system response speed in the idle state.

3.1. MTConnect Agent and MTConnect Adapter

The MTConnect adapter generates selected elements from the “DataItem” of the automated DNA extraction system’s information model, based on real-time hardware information from the automated DNA extraction system. This allows the MTConnect agent to respond to requests from MTConnect clients. The MTConnect agent works as a structure to prepare for data requests from clients by collecting target system data from the adapter at 100 ms intervals and storing them in a buffer.

There are three types of data that an MTConnect client can request from the MTConnect agent. “Prove” requests device components and data items, “Current” requests data item values or device status information at a specific point in time, and “Sample” requests samples, events, status information, etc. for a specific continuous period. These three types of data form the basis of the MTConnect standard, and they are implemented to respond to client requests correctly and accurately (Figure 8).

```

← → ↻ 🔍 localhost:1080/probe
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<?xml version="1.0" encoding="UTF-8" ?>
<MTConnectDevices xmlns="urn:mtconnect.org:MTConnectDevices:1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.2 ../MTConnectDevices:1.2.xsd">
  <Header assetBufferSize="10" assetCount="0" bufferSize="100" creationTime="2020-10-20T10:18:00" instanceId="1" sender="raspberrypi" version="1.6"/>
  <Devices>
    <Device id="LabGenius" name="labgenius.Digilog" uuid="3A200617H0">
      <DataItems>
        <DataItem category="EVENT" id="availability" name="availability" type="AVAILABILITY"/>
      </DataItems>
      <Components>
        <Axes id="axes" name="Axes">
          <Rotary id="x" name="Chamber">
            <DataItems>
              <DataItem category="EVENT" coordinateSystem="MACHINE" id="chamRdy" name="Chamber Ready State" type="EXECUTION"/>
              <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="chamVel" name="Actual Chamber RotrVelocity" subtype="ACTUAL" type="ROTARY_VELOCITY"/>
              <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="chamPos" name="Actual Chamber Position" subtype="ACTUAL" type="POSITION"/>
            </DataItems>
          </Rotary>
          <Linear id="y" name="Syringe">
            <DataItems>
              <DataItem category="EVENT" coordinateSystem="MACHINE" id="syrRdy" name="Syringe Ready State" type="EXECUTION"/>
              <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="syrVel" name="Actual Syringe Velocity" type="VELOCITY"/>
              <DataItem category="SAMPLE" coordinateSystem="MACHINE" id="syrPos" name="Actual Syringe Position" subtype="ACTUAL" type="POSITION"/>
            </DataItems>
          </Linear>
        </Axes>
        <Sensor id="platform" name="Platform">
          <DataItems>
            <DataItem category="SAMPLE" id="platformTemp" name="Platform Temp Sensor" type="TEMPERATURE"/>
            <DataItem category="EVENT" id="platformReady" name="Platform Ready State" type="EXECUTION"/>
          </DataItems>
        </Sensor>
        <Controller id="locomotion" name="LocoMotDriver">
          <Components>
            <Path id="Path" name="Path">
              <DataItems>
                <DataItem category="EVENT" id="powerError" name="Power Error" type="MESSAGE"/>
                <DataItem category="EVENT" id="waitForButton" name="Wait For Button" type="MESSAGE"/>
                <DataItem category="EVENT" id="heatShutdown" name="Heat Shutdown" type="MESSAGE"/>
                <DataItem category="EVENT" id="preheat" name="Preheat" type="MESSAGE"/>
                <DataItem category="EVENT" id="buildCancel" name="Build Canceling" type="MESSAGE"/>
              </DataItems>
            </Path>
          </Components>
        </Controller>
      </Components>
    </Device>
  </Devices>
</MTConnectDevices>

```

(a) Probe.

Figure 8. Cont.

his XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<MTConnectStreams xmlns="urn:mtconnect.org:MTConnectStreams:1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.2 http://www.mtconnect.org/schemas/MTConnectStreams_1.2.xsd">
  <Header bufferSize="100" creationTime="2020-10-20T10:18:00" firstSequence="3055" instanceId="1" lastSequence="3153" nextSequence="3154" sender="raspberrypi"
    version="1.6"/>
  <Streams>
    <DeviceStream name="labgenius_Digilog" uuid="3A200617HD">
      <ComponentStream component="Device" componentId="LabGenius" name="labgenius_Digilog">
        <Events>
          <Availability dataItemId="availability" name="availability" sequence="3153" timestamp="2020-11-12T03:49:51">AVAILABLE</Availability>
        </Events>
      </ComponentStream>
      <ComponentStream component="Rotary" componentId="x" name="Chamber">
        <Samples>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="3153" subType="ACTUAL" timestamp="2020-11-12T03:49:51">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="3153" subType="ACTUAL" timestamp="2020-11-12T03:49:51">0.0</Position>
        </Samples>
        <Events>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="3153" timestamp="2020-11-12T03:49:51">READY</Execution>
        </Events>
      </ComponentStream>
      <ComponentStream component="Linear" componentId="y" name="Syringe">
        <Samples>
          <Velocity dataItemId="syrVel" name="Actual Syringe Velocity" sequence="3153" timestamp="2020-11-12T03:49:51">13456</Velocity>
          <Position dataItemId="syrPos" name="Actual Syringe Position" sequence="3153" subType="ACTUAL" timestamp="2020-11-12T03:49:51">0.0</Position>
        </Samples>
        <Events>
          <Execution dataItemId="syrRdy" name="Syringe Ready State" sequence="3153" timestamp="2020-11-12T03:49:51">READY</Execution>
        </Events>
      </ComponentStream>
      <ComponentStream component="Sensor" componentId="platform" name="Platform">
        <Samples>
          <Temperature dataItemId="platformTemp" name="Platform Temp Sensor" sequence="3153" timestamp="2020-11-12T03:49:51">UNAVAILABLE</Temperature>
        </Samples>
        <Events>
          <Execution dataItemId="platformReady" name="Platform Ready State" sequence="3153" timestamp="2020-11-12T03:49:51">UNAVAILABLE</Execution>
        </Events>
      </ComponentStream>
      <ComponentStream component="Path" componentId="Path" name="Path">
        <Events>
          <Message dataItemId="powerError" name="Power Error" sequence="3153" timestamp="2020-11-12T03:49:51">UNAVAILABLE</Message>
          <Message dataItemId="waitForButton" name="Wait For Button" sequence="3153" timestamp="2020-11-12T03:49:51">UNAVAILABLE</Message>
          <Message dataItemId="heatShutdown" name="Heat Shutdown" sequence="3153" timestamp="2020-11-12T03:49:51">UNAVAILABLE</Message>
          <Message dataItemId="preheat" name="Preheat" sequence="3153" timestamp="2020-11-12T03:49:51">UNAVAILABLE</Message>
          <Message dataItemId="buildCancel" name="Build Canceling" sequence="3153" timestamp="2020-11-12T03:49:51">UNAVAILABLE</Message>
        </Events>
      </ComponentStream>
    </DeviceStream>
  </Streams>
</MTConnectStreams>
```

(b) Current.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<MTConnectStreams xmlns="urn:mtconnect.org:MTConnectStreams:1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mtconnect.org:MTConnectStreams:1.2 http://www.mtconnect.org/schemas/MTConnectStreams_1.2.xsd">
  <Header bufferSize="10" creationTime="2020-10-20T10:18:00" firstSequence="359" instanceId="1" lastSequence="367" nextSequence="368" sender="raspberrypi"
    version="1.6"/>
  <Streams>
    <DeviceStream name="labgenius_Digilog" uuid="3A200617HD">
      <ComponentStream component="Device" componentId="LabGenius" name="labgenius_Digilog">
        <Events>
          <Availability dataItemId="availability" name="availability" sequence="359" timestamp="2020-11-12T03:37:29">AVAILABLE</Availability>
          <Availability dataItemId="availability" name="availability" sequence="360" timestamp="2020-11-12T03:37:29">AVAILABLE</Availability>
          <Availability dataItemId="availability" name="availability" sequence="361" timestamp="2020-11-12T03:37:29">AVAILABLE</Availability>
          <Availability dataItemId="availability" name="availability" sequence="362" timestamp="2020-11-12T03:37:29">AVAILABLE</Availability>
          <Availability dataItemId="availability" name="availability" sequence="363" timestamp="2020-11-12T03:37:29">AVAILABLE</Availability>
          <Availability dataItemId="availability" name="availability" sequence="364" timestamp="2020-11-12T03:37:29">AVAILABLE</Availability>
          <Availability dataItemId="availability" name="availability" sequence="365" timestamp="2020-11-12T03:37:30">AVAILABLE</Availability>
          <Availability dataItemId="availability" name="availability" sequence="366" timestamp="2020-11-12T03:37:30">AVAILABLE</Availability>
          <Availability dataItemId="availability" name="availability" sequence="367" timestamp="2020-11-12T03:37:30">AVAILABLE</Availability>
        </Events>
      </ComponentStream>
      <ComponentStream component="Rotary" componentId="x" name="Chamber">
        <Samples>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="359" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="359" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Position>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="360" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="360" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Position>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="361" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="361" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Position>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="362" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="362" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Position>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="363" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="363" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Position>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="364" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="364" subType="ACTUAL" timestamp="2020-11-12T03:37:29">0.0</Position>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="365" subType="ACTUAL" timestamp="2020-11-12T03:37:30">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="365" subType="ACTUAL" timestamp="2020-11-12T03:37:30">0.0</Position>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="366" subType="ACTUAL" timestamp="2020-11-12T03:37:30">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="366" subType="ACTUAL" timestamp="2020-11-12T03:37:30">0.0</Position>
          <Rotary_velocity dataItemId="chamVel" name="Actual Chamber RotrVelocity" sequence="367" subType="ACTUAL" timestamp="2020-11-12T03:37:30">0.0</Rotary_velocity>
          <Position dataItemId="chamPos" name="Actual Chamber Position" sequence="367" subType="ACTUAL" timestamp="2020-11-12T03:37:30">0.0</Position>
        </Samples>
        <Events>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="359" timestamp="2020-11-12T03:37:29">READY</Execution>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="360" timestamp="2020-11-12T03:37:29">READY</Execution>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="361" timestamp="2020-11-12T03:37:29">READY</Execution>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="362" timestamp="2020-11-12T03:37:29">READY</Execution>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="363" timestamp="2020-11-12T03:37:29">READY</Execution>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="364" timestamp="2020-11-12T03:37:29">READY</Execution>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="365" timestamp="2020-11-12T03:37:30">READY</Execution>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="366" timestamp="2020-11-12T03:37:30">READY</Execution>
          <Execution dataItemId="chamRdy" name="Chamber Ready State" sequence="367" timestamp="2020-11-12T03:37:30">READY</Execution>
        </Events>
      </ComponentStream>
    </DeviceStream>
  </Streams>
```

(c) Sample.

Figure 8. MTConnect agent-generated XMLs.

3.2. MTConnect Monitoring Client Application

The MTConnect monitoring client application is located on the same page as the web application-based user control UI (Figure 9), so that user control feedback can be immediately checked through MTConnect monitoring, enhancing the usefulness of function utilization. In addition, by implementing a web application-based client using Vue.JS, we sought to guarantee the same user experience regardless of the client's platform and equipment.

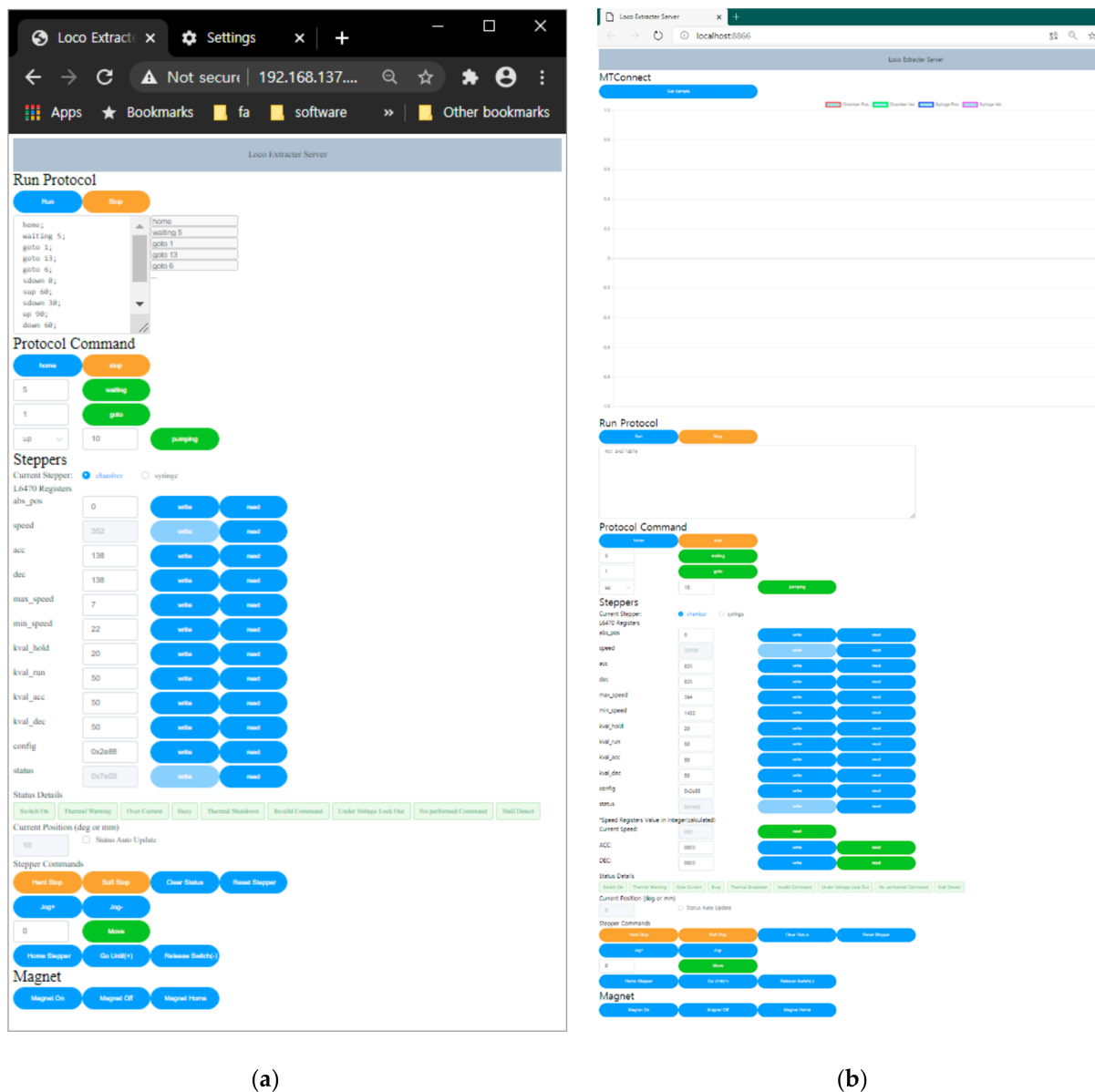
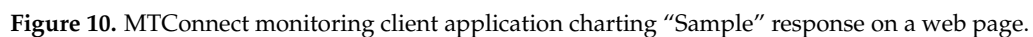


Figure 9. (a) Web application of DNA extraction system control; (b) MTConnect monitoring client web application.

When the MTConnect monitoring client application clicks the “Get Sample” button on the UI, it sends a “Sample” request to the MTConnect agent. After receiving the response data, the speed and position elements of each motor drive shaft are extracted using XML Parser. This was implemented to create a chart on a web page using the responsive chart library “vue-charts” (Figure 10).



```

57 | ..... <Sensor id="platform" name="Platform">
58 | ..... <DataItems>
59 | ..... <DataItem type="TEMPERATURE" id="platformTemp" category="SAMPLE" name="Platform Temp Sensor"/>
60 | ..... <DataItem type="EXECUTION" id="platformReady" category="EVENT" name="Platform Ready State" />
61 | ..... </DataItems>
62 | ..... </Sensor> -->
63 | .....
64 | ..... <Controller id="locomotion" name="locoMotDriver">
65 | ..... <Components>
66 | ..... <Path id="..... Path" name="..... Path">
67 | ..... <DataItems>
68 | ..... <!-- "syrgOcd":"UNAVAILABLE", "syrgThwrn":"UNAVAILABLE", "syrgStalldet":"UNAVAILABLE", -->
69 | ..... <DataItem type="MESSAGE" id="syrgOcd" category="EVENT" name="Syringe Over Current Detection"/>
70 | ..... <DataItem type="MESSAGE" id="syrgThwrn" category="EVENT" name="Syringe Thermal Warning"/>
71 | ..... <DataItem type="MESSAGE" id="syrgStalldet" category="EVENT" name="Syringe Stall Detection"/>
72 | .....
73 | ..... <DataItem type="MESSAGE" id="powerError" category="EVENT" name="Power Error"/>
74 | ..... <DataItem type="MESSAGE" id="waitForButton" category="EVENT" name="Wait For Button"/>
75 | ..... <DataItem type="MESSAGE" id="heatShutdown" category="EVENT" name="Heat Shutdown"/>
76 | ..... <DataItem type="MESSAGE" id="preheat" category="EVENT" name="Preheat"/>
77 | ..... <DataItem type="MESSAGE" id="buildCancel" category="EVENT" name="Build Canceling"/>
78 | ..... </DataItems>
79 | ..... </Path>
80 | ..... </Components>
81 | ..... </Controller>
82 | .....
83 | ..... </Components>
84 | ..... </Device>
85 |

```

Figure 11. MTConnect agent monitoring major failure factors.

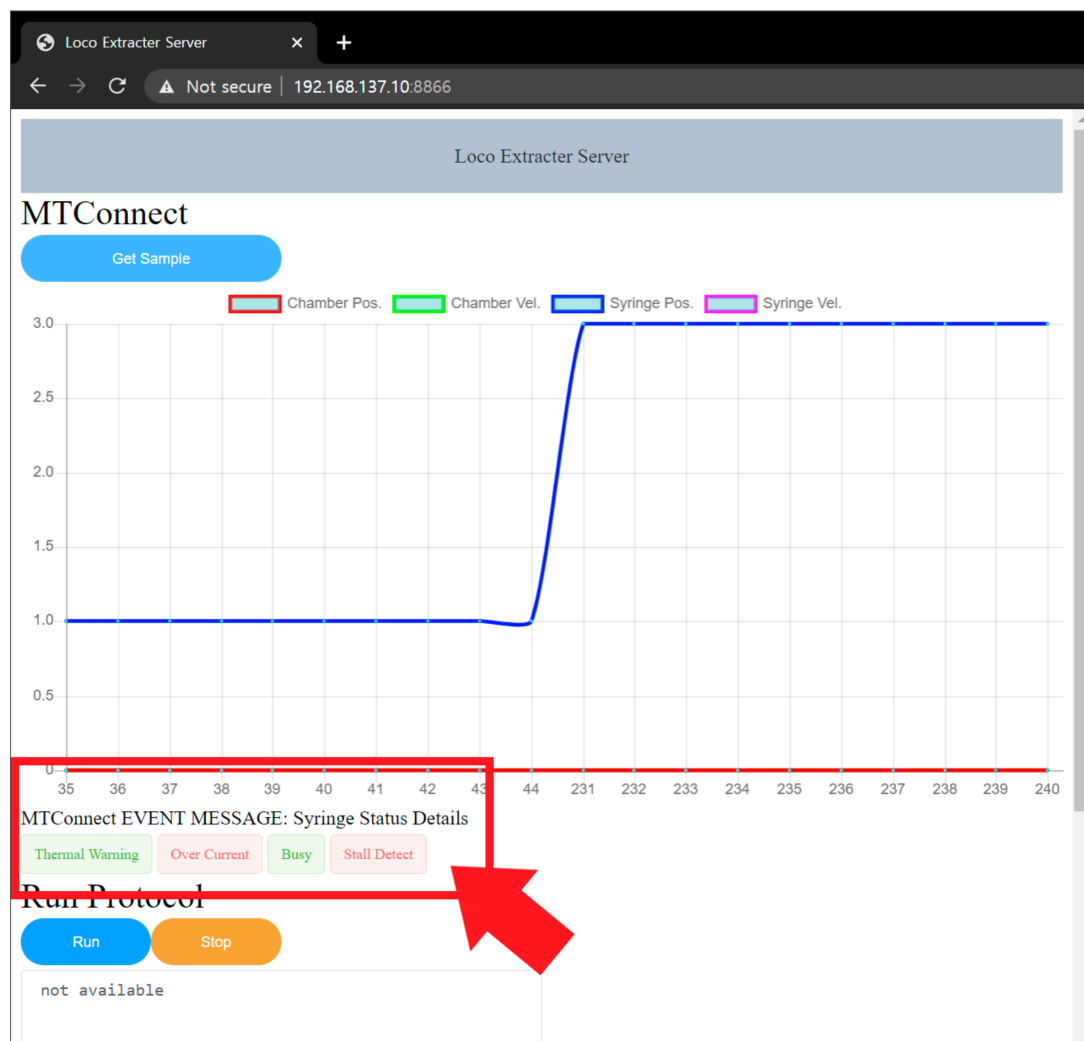


Figure 12. MTConnect client web application with failure message.

3.3. Performance Evaluation

The performance and reliability of the entire system were evaluated by verifying the request response time for each section of the data flow from the web server of the implemented system to the device driver. Table 1 shows statistics obtained by repeatedly measuring the request response time for each section 1000 times when the automated DNA extraction system is in the idle state.

Table 1. Average response time of the implemented system.

| Data Flow | Web Server ↔ MTConnect Agent | | | MTConnect Agent ↔ MTConnect Adapter | MTConnect Adapter ↔ POCT Device |
|----------------|------------------------------|----------|----------|-------------------------------------|---------------------------------|
| Request | /Sample = 100 | /Probe | /Current | pollDevice() | chamber_status |
| No. of Char. ≈ | ≈180,500 | ≈3600 | ≈3000 | ≈500 | ≈250 |
| Average (sec) | 0.011390 | 0.009620 | 0.010551 | 0.034464 | 0.015726 |
| STDEV | 0.003846 | 0.002988 | 0.003944 | 0.004110 | 0.002533 |
| Min. (sec) | 0.004416 | 0.004351 | 0.004338 | 0.023927 | 0.010632 |
| Max. (sec) | 0.026729 | 0.023792 | 0.032317 | 0.059352 | 0.047276 |

When the request response time of the measurement target section is measured, the time required to execute the method is saved as a log file, and the data are obtained and

analyzed. As a result of reviewing the data obtained after several repeated tests for a long period of time, there was no error in the method execution result or number, so we came to the conclusion that the system described in this study operates very stably.

4. Discussion

POCT is used in various fields, such as medicine, quarantine, agriculture (breeding), and food. In particular, its use in the medical field is very important because it takes a short time from the start of a test to diagnosis and is not restricted by location and time. The reason why this is receiving such attention from the industry is that the patient's condition changes from moment to moment, and the appropriate response at the beginning is effective for treatment. In addition, the necessity of being able to monitor POCT equipment and manage it remotely is particularly important for the specialty of remote medical treatment.

This study demonstrated the possibility of the remote monitoring of automated diagnosis processes using POCT devices and the MTConnect standard technology, in the field of in vitro diagnostic testing. Previous studies on the MTConnect standard technology were limited in application to equipment monitoring in the manufacturing industry. This study has demonstrated that MTConnect standard technology can be applied to monitoring the status of medical equipment in multiple remote locations in real time.

The average response time of the system implemented in this study was longer (11.39 ms, 9.62 ms, and 10.55 ms for sample, probe, and current commands, respectively) than that of the system described in a previous MTConnect study (4.80 ms, 4.81 ms, and 4.50 ms for sample, probe, and current commands, respectively) [21]. The reason for the longer response time of this study seems to be that the proposed system handles not only the MTConnect function but also the DNA extraction process. In this study, it was expected that the optimization of each program implemented on a single-board computer and running at the same time was necessary, along with optimization of the interlocking parts.

In a follow-up study, the adaptability and performance of the currently implemented MTConnect monitoring solution to other medical devices must be tested. This will require efforts to improve the performance of the MTConnect system and expand its application to other companies' equipment in the manufacturing facility industry.

5. Conclusions

The architecture and implementation of a web application-based monitoring system for automated DNA extraction using MTConnect were examined in this study. A single-board computer was designed to measure the performance of the system and test its stability and reliability. The results show that the remote monitoring of POCT equipment is possible using the proposed system. In addition to improving the performance of the system through additional optimization efforts in the future, we plan to expand the system's application to other types of medical equipment and to equipment used in other fields.

Author Contributions: Conceptualization, A.-J.P.; methodology, M.-H.R.; software, S.-H.H. and M.-H.R.; validation, A.-R.P.; data curation, S.-H.H.; writing—original draft preparation, S.-H.H.; writing—review and editing, M.-H.R.; visualization, A.-R.P.; supervision, A.-R.P.; project administration, M.-H.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Korea Medical Device Development Fund, Project Number: HW20C2053.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data not publicly available.

Acknowledgments: This work was supported by the Korea Medical Device Development Fund grant funded by the Korean government (Project Number: HW20C2053).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, W.-C.; Nguyen, M.-H.; Tai, P.-H. An Intelligent Manufacturing System for Injection Molding. *Proc. Eng. Technol. Innov.* **2018**, *8*, 9.
- Lee, W.; Shin, S. An empirical study of consumer adoption of Internet of Things services. *Int. J. Eng. Technol. Innov.* **2019**, *9*, 1.
- MTConnect. Available online: <http://www.mtconnect.org> (accessed on 20 August 2020).
- Alhama Blanco, P.J.; Abu-Dakka, F.J.; Abderrahim, M. Practical use of robot manipulators as intelligent manufacturing systems. *Sensors* **2018**, *18*, 2877. [[CrossRef](#)] [[PubMed](#)]
- Zabalza, J.; Fei, Z.; Wong, C.; Yan, Y.; Mineo, C.; Yang, E.; Rodden, T.; Mehnen, J.; Pham, Q.-C.; Ren, J. Smart sensing and adaptive reasoning for enabling industrial robots with interactive human-robot capabilities in dynamic environments—A case study. *Sensors* **2019**, *19*, 1354. [[CrossRef](#)] [[PubMed](#)]
- Foumani, M.; Gunawan, I.; Smith-Miles, K. Increasing throughput for a class of two-machine robotic cells served by a multifunction robot. *IEEE Trans. Autom. Sci. Eng.* **2015**, *14*, 1150–1159. [[CrossRef](#)]
- Edrington, B.; Zhao, B.; Hansel, A.; Mori, M.; Fujishima, M. Machine monitoring system based on MTConnect technology. *Procedia Cirp* **2014**, *22*, 92–97. [[CrossRef](#)]
- Shin, S.-J.; Woo, J.; Seo, W.; Jeong, Y.-J. Energy prediction modeling for numerical control programs using MTconnect. *J. Korean Soc. Precis. Eng.* **2017**, *34*, 355–362. [[CrossRef](#)]
- Hu, L.; Nguyen, N.-T.; Tao, W.; Leu, M.C.; Liu, X.F.; Shahriar, M.R.; Al Sunny, S.N. Modeling of cloud-based digital twins for smart manufacturing with MT connect. *Procedia Manuf.* **2018**, *26*, 1193–1203. [[CrossRef](#)]
- KISTI. Available online: <https://www.ntis.go.kr> (accessed on 15 October 2020).
- KHIDI. Available online: <https://www.khidi.or.kr> (accessed on 15 October 2020).
- TIPA. Available online: <https://www.tipa.or.kr> (accessed on 15 October 2020).
- NIFDS. Available online: <https://www.nifds.go.kr> (accessed on 15 October 2020).
- Christodouleas, D.C.; Kaur, B.; Chorti, P. From point-of-care testing to eHealth diagnostic devices (eDiagnostics). *ACS Cent. Sci.* **2018**, *4*, 1600–1616. [[CrossRef](#)] [[PubMed](#)]
- Tsakalakis, M.; Bourbakis, N.G. Health care sensor—Based systems for point of care monitoring and diagnostic applications: A brief survey. In Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, USA, 26–30 August 2014; pp. 6266–6269.
- MTConnect Institute. MTConnect®Standard Part 1—Overview and Fundamentals. Available online: https://docs.mtconnect.org/MTC_Part_1-0-Overview_and_Fundamentals_1-6-0.pdf (accessed on 20 August 2020).
- Kim, K.-W.; Lee, M.-S.; Ryu, M.-H.; Kim, J.-W. Arduino-based automation of a DNA extraction system. *Technol. Health Care* **2016**, *24*, S105–S112. [[CrossRef](#)] [[PubMed](#)]
- MTConnect Institute. Getting Started with MTConnect—Connectivity Guide. Available online: <https://mtconnect.squarespace.com/s/Getting-Started-with-MTConnect-FINAL.pdf> (accessed on 20 August 2020).
- Vijayaraghavan, A.; Huet, L.; Dornfeld, D.; Sobel, W. *Addressing Process Planning and Verification Issues with MTConnect*. Trans. 37; North American Research Institute: San Dimas, CA, USA, 2009.
- Makerbot_Agent. Available online: https://github.com/mtconnect/makerbot_agent (accessed on 20 August 2020).
- Liu, X.F.; Sunny, S.N.A.; Shahriar, M.R.; Leu, M.C.; Cheng, M.; Hu, L. *Implementation of MTConnect for Open Source 3D Printers in Cyber Physical Manufacturing Cloud, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*; American Society of Mechanical Engineers: New York, NY, USA, 2016; p. V01AT02A019.