

การศึกษาฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์

A Survey of Kria KR260 Hardware Acceleration for Robotics

อภิรักษ์ รัตนฤมิต

สาขาวิชาวิศวกรรมหุ่นยนต์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสวนสุนันทา

1 ถนนอุทงนอก เขตดุสิต กทม. 10300

E-mail: aphirak.th@ssru.ac.th

เศรษฐกาล โปร่งนุช

สาขาวิชาวิศวกรรมหุ่นยนต์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสวนสุนันทา

1 ถนนอุทงนอก เขตดุสิต กทม. 10300

E-mail: sethakarn.pr@ssru.ac.th

ปริญญญาณ์ แสงอรุณ

สาขาวิชาการออกแบบผลิตภัณฑ์และบรรจุภัณฑ์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสวนสุนันทา

1 ถนนอุทงนอก เขตดุสิต กทม. 10300

E-mail: patinya.sa@ssru.ac.th

Manuscript received November 30, 2022

Revised December 19, 2022

บทคัดย่อ

บทความนี้นำเสนอการศึกษาฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์ เป็นแพลตฟอร์มการพัฒนาที่ใช้ Kria System-on-Modules (SOM) ใช้สำหรับการพัฒนาหุ่นยนต์และระบบอัตโนมัติในโรงงาน ช่วยให้วิศวกรหุ่นยนต์และนักพัฒนาอุตสาหกรรมที่ไม่มีความเชี่ยวชาญด้านเอฟพีจีเอ (FPGA) สามารถพัฒนาแอปพลิเคชันบนฮาร์ดแวร์ประมวลผล การประมวลผลภาพและวิดีโอ การสื่อสารและการควบคุมในอุตสาหกรรม นักพัฒนาสามารถใช้ประโยชน์จากระบบปฏิบัติการสำหรับหุ่นยนต์รุ่นที่ 2 Robot Operating System 2 (ROS 2) มีประสิทธิภาพการทำงานที่เพิ่มขึ้นเมื่อทำงานร่วมกับ Kria Robotics Stack (KRS) อินเทอร์เน็ตและแอปพลิเคชันทำให้ KR260 เป็นแพลตฟอร์มในนวัตกรรมหุ่นยนต์ระดับอุตสาหกรรม ด้วย ROS2 ซึ่งทำให้ฮาร์ดแวร์ที่ใช้เอฟพีจีเอ

ทำงานแบบเรียลไทม์ได้อย่างมีประสิทธิภาพ และงานที่ต้องการด้านความปลอดภัย ทำให้ Kria KR260 เป็นตัวเลือกใหม่สำหรับนักวิทยาการหุ่นยนต์และนักพัฒนาอุตสาหกรรม

คำสำคัญ: คีร์รา เคอาร์260 สแต็คหุ่นยนต์คีร์รา ระบบปฏิบัติการหุ่นยนต์รุ่นที่ 2 บอร์ดพัฒนาซิลิก

ABSTRACT

This paper presents a study of the Kria KR260 processing hardware for robots. It is a development platform based on Kria SOM for robotics and factory automation. It enables robotics and industrial developers without FPGA experience to develop applications with processing hardware, image and video processing, industrial communication and control. Developers can take advantage of the Robot Operating System 2 (ROS 2) 2nd generation robotic operating

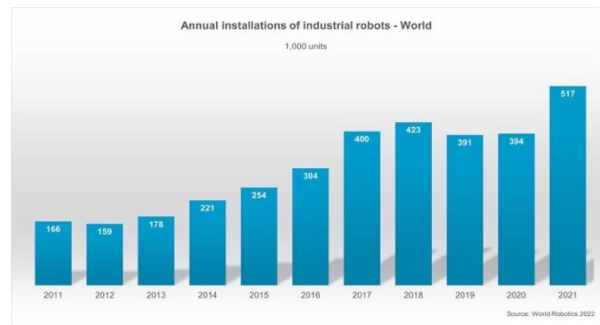
system and enhanced performance through the Kria Robotics Stack (KRS). It is an innovative industrial robot platform with ROS2, which enables real-time performance of FPGA-based hardware. And an emphasis on safety architecture makes the Kria KR260 the new choice for robotics and industrial developers.

Keywords: Kria KR260, Kria Robotics Stack (KRS), Robot Operating System 2, Xilinx development board.

1. บทนำ

ความต้องการหุ่นยนต์กำลังขยายตัวขึ้นอย่างรวดเร็ว จากข้อมูลของบริษัทวิจัย Statista พบว่ามูลค่าของตลาดทั่วโลกสำหรับหุ่นยนต์อุตสาหกรรมจะเพิ่มขึ้นจาก 81 พันล้านดอลลาร์ในปี พ.ศ. 2564 เป็นมากกว่า 165 พันล้านดอลลาร์ในปี พ.ศ. 2571 [1] ปัจจุบันเทคโนโลยีที่จำเป็นในการสร้างหุ่นยนต์ที่ปลอดภัยและสามารถทำงานควบคู่ไปกับมนุษย์ได้ แต่การใช้เทคโนโลยีเหล่านี้ทำงานร่วมกันอาจเป็นงานที่ซับซ้อน คือ การเพิ่มปัญญาประดิษฐ์ในการคิดคำนวณ ดังนั้นเพื่อตอบสนองนวัตกรรมที่รวดเร็วในปัจจุบัน นักวิทยาการหุ่นยนต์จึงหันมาใช้แพลตฟอร์มการคำนวณแบบปรับตัวได้ ช่วยลดเวลาหน่วง (Low latency) ให้ต่ำลงและสามารถกำหนดการควบคุมหุ่นยนต์แบบหลายแกน ได้พร้อมระบบรักษาความปลอดภัยในตัวบนแพลตฟอร์มโมดูลาร์ที่สามารถปรับสเกลได้ในอนาคต [1]

ในปี ค.ศ. 2021 มีรายงานจาก World Robotics ระบุว่า มีหุ่นยนต์เกิดขึ้นอุตสาหกรรมใหม่จำนวน 517,385 ตัว ถูกติดตั้งในโรงงานต่าง ๆ ทั่วโลกสูงสุดเป็นประวัติการณ์ แสดงอัตราการเติบโตที่ 31% เมื่อเทียบเป็นรายปี และสูงกว่าสถิติการติดตั้งหุ่นยนต์ก่อนเกิดโรคระบาดในปี ค.ศ. 2018 ถึง 22% ปัจจุบันนี้คลังเก็บของหุ่นยนต์สำหรับปฏิบัติการทั่วโลกทำสถิติใหม่ประมาณ 3.5 ล้านหน่วย [2] เอเชียยังคงเป็นตลาดหุ่นยนต์อุตสาหกรรมที่ใหญ่ที่สุดในโลกคิดเป็น 74% ของหุ่นยนต์ทั้งหมด ในปี ค.ศ. 2020 - 2021 และหุ่นยนต์ถูกติดตั้งในเอเชียไปถึง 70%



รูปที่ 1 การติดตั้งหุ่นยนต์ประจำปีทั่วโลกระหว่างปี 2015 ถึง 2021 เพิ่มขึ้นกว่าเท่าตัว [2]

จากรูปที่ 1 การติดตั้งหุ่นยนต์ประจำปีทั่วโลกระหว่างปี ค.ศ. 2015 ถึง ค.ศ. 2021 เพิ่มขึ้นกว่าเท่าตัว ซึ่งหุ่นยนต์ดังกล่าวเป็นการผสมผสานระหว่างเทคโนโลยีที่หลากหลายรวมกันเป็นหนึ่งเดียว รวมถึงการควบคุมอุตสาหกรรมและการสื่อสาร วิทยาศาสตร์ การเรียนรู้ของเครื่อง (Machine Learning) ปัญญาประดิษฐ์ ส่วนต่อประสานผู้ใช้ที่เชื่อมต่องานกับตัวควบคุมสำหรับระบบอุตสาหกรรม (Human Machine Interface: HMI) ระบบความปลอดภัย เป็นต้น [1] หุ่นยนต์ได้รับการออกแบบมาเพื่อทำงานเฉพาะทาง จึงต้องการระบบสมองกลฝังตัวที่มีระดับความฉลาดมากขึ้น (โปรเซสเซอร์ที่มีประสิทธิภาพในการคำนวณได้มากกว่าการคำนวณปกติ) และความสามารถในการรองรับอินพุตเซ็นเซอร์ต่าง ๆ ต้องใช้ปัญญาประดิษฐ์บางรูปแบบสำหรับการวิเคราะห์แบบเรียลไทม์ การบำรุงรักษาเชิงคาดการณ์ (Predictive Maintenance) การวินิจฉัยระยะไกล และอื่น ๆ

หุ่นยนต์อุตสาหกรรมและการแพทย์ระดับมืออาชีพในปัจจุบันจำนวนมาก มีเทคโนโลยีหลักสองอย่าง เพื่อใช้ในการขับเคลื่อนคือ 1) ตัวประมวลผลเพื่อจัดการข้อมูลที่ซับซ้อนและโครงสร้างการควบคุมที่สร้างกราฟการคำนวณ และ 2) เอสโอซีแบบปรับตัวได้ (Adaptive SoC) ที่ใช้ในการรับสัญญาณ ประมวลผลในแบบเรียลไทม์และโอนไปยังตัวประมวลผล เพื่อดำเนินการต่อไป [1] เอสโอซีแบบปรับตัวได้สามารถช่วยได้สามวิธี: 1) การลดภาระในการคำนวณผลที่สำคัญต่อเวลา 2) การเร่งฟังก์ชันการคำนวณบนฮาร์ดแวร์เพื่อรักษาความสมดุลระหว่างการคำนวณและเวลาตอบสนอง 3) ลดพลังงานโดยรวมในการดำเนินการคำนวณแบบคู่ขนาน

บทความนี้จะนำเสนอ ข้อมูลต่าง ๆ จากการศึกษาฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์จากงานวิจัยต่างๆ การประยุกต์ใช้งาน มีการเปรียบเทียบคุณลักษณะความสามารถในด้านต่างๆ และแนวทางในการพัฒนาเพื่อใช้งาน โดยในหัวข้อถัดไปจะกล่าวถึงการทฤษฎีที่เกี่ยวข้อง ฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์ แอปพลิเคชันบนฮาร์ดแวร์ประมวลผล และบทสรุปตามลำดับ

2. ทฤษฎีที่เกี่ยวข้อง

การศึกษาฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์ในบทความนี้ แบ่งออกเป็น 2 ส่วน ได้แก่ งานวิจัยทางด้านเทคโนโลยีประมวลผลสำหรับหุ่นยนต์ และงานวิจัยทางด้านระบบสมองกลฝังตัวและเอสไอซี

2.1 เทคโนโลยีประมวลผลสำหรับหุ่นยนต์

นักวิทยาการหุ่นยนต์สามารถทำงานร่วมกับ FPGA ได้สามวิธี [1]:

- 1) วิธีการชิปดาวน์ (Chip-down approach)

เมื่อเอสไอซี (SoC) ถูกรวมเข้ากับ PCB ที่ออกแบบไว้เพื่อตอบสนองความต้องการของแอปพลิเคชัน สำหรับการสร้างวงจรขนาดใหญ่ที่มีความคุ้มค่า

- 2) ระบบบนโมดูล (SOM)

บอร์ดสำเร็จรูปที่ประกอบและผ่านการทดสอบแล้ว เชื่อมต่อเข้ากับบอร์ดแบบกำหนดเอง และช่วยให้วิศวกรสร้างผลิตภัณฑ์ของตนได้เร็วยิ่งขึ้น โดยมุ่งเน้นไปที่การเพิ่มมูลค่าให้กับระบบ แทนที่จะมุ่งไปที่การผสานรวม การทดสอบ และการรับรอง

- 3) บอร์ดประกอบที่สมบูรณ์ (Fully assembled board)

ในกรณีนี้ อุปกรณ์ต่อพ่วงจำนวนมากถูกรวมในบอร์ดหลักอื่น มีความเหมาะสมอย่างยิ่งสำหรับการใช้งานที่มีการประมวลผลสูง

อีกรูปแบบหนึ่งของการประมวลผลที่ใช้ในวิทยาการหุ่นยนต์คือการคำนวณแบบปรับตัว การประมวลผลแบบปรับตัวช่วยให้เร่งความเร็วด้วยฮาร์ดแวร์ ทำให้การประมวลผลเร็วขึ้น ลดการใช้พลังงาน และกำหนดพฤติกรรมได้มากขึ้น ด้วยเครื่องมือเร่งความเร็วที่เหมาะสม หุ่นยนต์สามารถออกแบบสถาปัตยกรรมการคำนวณที่ปรับทรัพยากรฮาร์ดแวร์ให้เหมาะสมสำหรับแอปพลิเคชันของตน

2.2 ระบบสมองกลฝังตัวและเอสไอซี

ระบบสมองกลฝังตัวและเอสไอซีมีการใช้งานในด้านวิศวกรรมจำนวนมากในปัจจุบัน [3] – [5] เอสไอซีมีวิธีการปรับปรุงการกำหนดและคาดการณ์ การคำนวณแบบปรับตัวให้ความสามารถพิเศษที่จำเป็นในการเคลื่อนที่หุ่นยนต์ไปสู่ความเป็นอิสระ นอกจากนี้ จะช่วยให้การพัฒนาระบบหุ่นยนต์โมดูลาร์ที่ปรับขนาดได้เร็วและมีประสิทธิภาพมากขึ้นแล้ว ยังช่วยเร่งการเติบโตของวิทยาการหุ่นยนต์ การประมวลผลแบบปรับตัวยังทำให้มีความสามารถในการประมวลผลเพิ่มเติมสำหรับปัญญาประดิษฐ์และการประมวลผลสัญญาณดิจิทัล พร้อมด้วยแบนด์วิดท์ข้อมูลขนาดใหญ่ที่จำเป็นในการรับมือกับข้อมูลจำนวนมากที่หุ่นยนต์ต้องประมวลผล [1]

การประมวลผลแบบปรับตัว (Adaptive computing) รวมโมดูลการทำงานต่าง ๆ เช่น ซีพียูแบบมัลติคอร์ ซึ่งจัดอยู่ในคลัสเตอร์ของโปรเซสเซอร์แบบเรียลไทม์และแอปพลิเคชันที่ได้รับการปรับให้เหมาะสมที่สุด พร้อมด้วยลอจิกที่ตั้งโปรแกรมได้ ตัวประมวลผลแบบตายาย และเอ็นจินอัจฉริยะ ทำให้สามารถกระจายภาระงานของหุ่นยนต์ไปยังสถาปัตยกรรมที่ดีที่สุดได้ พลัสในการคำนวณทั้งหมดนั้นเสริมด้วยความสามารถด้านความปลอดภัยในการทำงานที่ทำให้หุ่นยนต์ปลอดภัยยิ่งขึ้น เพื่อหลีกเลี่ยงการละเมิดที่อาจส่งผลกระทบต่อความสมบูรณ์ของหุ่นยนต์และทำให้มันเสี่ยง หุ่นยนต์เป็นการผสมของการควบคุมและการตอบสนองต่อข้อมูลจากเหตุการณ์ภายนอก เช่น การกระตุ้นด้วยภาพหรือเซ็นเซอร์ การคำนวณแบบปรับตัวช่วยให้สามารถกำหนดการคำนวณที่เหมาะสมให้กับปริมาณงานทางคอมพิวเตอร์ที่เหมาะสม เมื่อรวมกับฮาร์ดแวร์แล้ว การออกแบบแพลตฟอร์มสำหรับการสร้างระบบที่มีความยืดหยุ่นสูงและมีประสิทธิภาพสูง [1]

3. ฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์

3.1 สถาปัตยกรรมฮาร์ดแวร์ประมวลผล Kria KR260

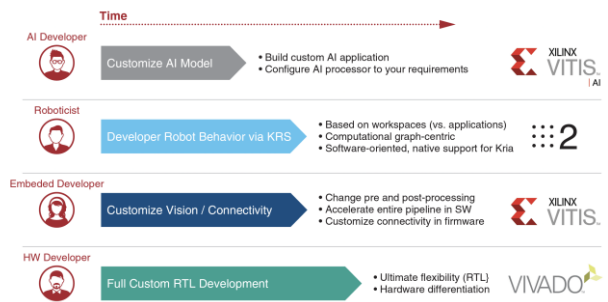
Kria KR260 Robotics Starter Kit เป็นแพลตฟอร์มการพัฒนาบนพื้นฐาน Kria SOM สำหรับหุ่นยนต์และระบบอัตโนมัติในโรงงาน ช่วยให้นักหุ่นยนต์ที่ไม่มีความเชี่ยวชาญด้านเอฟพีจีเอสามารถพัฒนาแอปพลิเคชัน การสื่อสารและการควบคุมในอุตสาหกรรม อีกทั้งยังได้ประโยชน์จากความยืดหยุ่นที่มากขึ้นจาก Robot Operating System 2 (ROS 2) และเพิ่มประสิทธิภาพการทำงานผ่าน Kria Robotics Stack (KRS)

3.2 เครื่องมือการพัฒนาสำหรับฮาร์ดแวร์ประมวลผล Kria KR260

นักพัฒนาแต่ละคนมีตัวเลือกเส้นทางการออกแบบที่สร้างขึ้น[6] สำหรับนักพัฒนาโปรแกรมตามที่ระบุในรูปที่ 2 นักพัฒนา AI สามารถวางโมเดล AI ที่กำหนดเองและผ่านการฝึกอบรมได้โดยใช้ Caffe, Tensorflow, Pytorch หรือเฟรมเวิร์กยอดนิยม ลงในตัวประมวลผล AI ที่ปรับเปลี่ยนได้

ตามข้อกำหนดด้านการออกแบบโดยการพัฒนารูปแบบได้ด้วยโปรแกรม Vitis AI

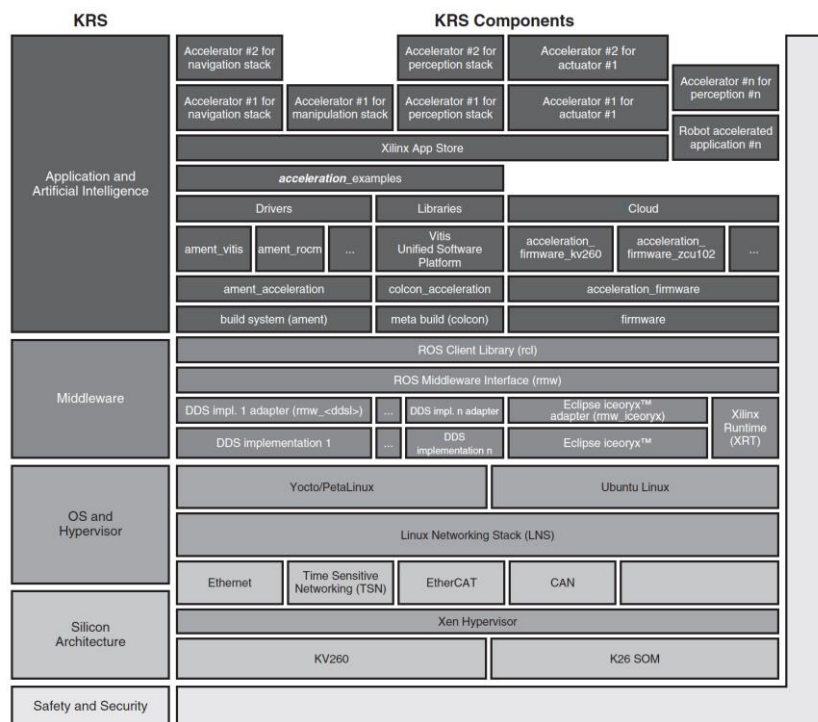
สำหรับนักพัฒนาวิทยาการด้านหุ่นยนต์ยังสามารถใช้ประโยชน์จาก KRS ซึ่งช่วยให้สร้างพื้นที่ทำงานสำหรับแพลตฟอร์ม Kria โดยใช้ ROS 2 นักพัฒนาตัวประมวลผลฝังตัวสามารถพัฒนาด้วย C++, OpenCL, Python และฮาร์ดแวร์ นักพัฒนาสามารถใช้ RTL เพื่อปรับแต่งระดับที่สูงขึ้นได้ตามต้องการโดยใช้เครื่องมือ Xilinx เช่น Vitis และ Vivado ตามลำดับ



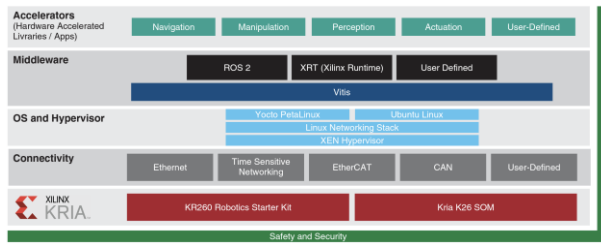
รูปที่ 2 Design Paths for Developers [6]

3.3 ระบบปฏิบัติการฮาร์ดแวร์ประมวลผล Kria KR260

KRS เป็น ROS 2 superset สำหรับอุตสาหกรรม รูปที่ 3-4 นำเสนอเครื่องมือและไลบรารีหุ่นยนต์ที่ช่วยสร้างโซลูชันหุ่นยนต์ที่กำหนดโดยซอฟต์แวร์ เป็นสะพานเชื่อมระหว่างหุ่นยนต์และ Xilinx ทำให้ใช้งานฮาร์ดแวร์สำหรับประมวลผลได้ง่ายขึ้น โดย KRS แบ่งส่วนประกอบออกเป็นห้ากลุ่ม: แอปพลิเคชันและปัญหาประดิษฐ์ มิดเดิลแวร์ (Middleware) ระบบปฏิบัติการและไฮเปอร์ไวเซอร์ (Hypervisor) สถาปัตยกรรมซิลิคอน ความปลอดภัย



รูปที่ 3 Kria Robotics Stack Tools and Components [7]

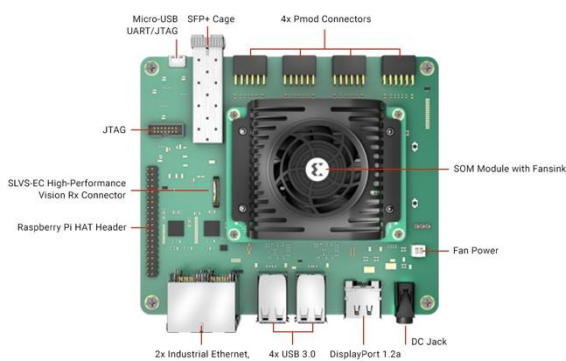


รูปที่ 4 Kria Robotics Stack (Simplified View) [6]

สำหรับนักวิทยาการหุ่นยนต์ ระบบสร้าง ROS 2 (ament) และเครื่องมือสร้าง ROS 2 (colcon) ที่เสริมศักยภาพและอำนวยความสะดวกใน กระบวนการสร้างคอร์เนลเร่งความเร็วจึงง่ายขึ้น ด้วยเครื่องมือเพิ่มประสิทธิภาพการทำงานที่สำคัญเหล่านี้ หุ่นยนต์ที่ใช้ ROS 2 สามารถสร้างและปรับใช้ฟังก์ชันด้วยฮาร์ดแวร์ได้อย่างรวดเร็ว

4. การติดตั้งระบบปฏิบัติการลินุกซ์บนฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์เบื้องต้น

ฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์สามารถติดตั้งระบบปฏิบัติการลินุกซ์บนฮาร์ดแวร์ในรูปที่ 5 โดยมีขั้นตอนดังนี้



รูปที่ 5 ฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์เบื้องต้น [6]

4.1 การติดตั้งไฟล์อิมเมจของระบบปฏิบัติการบนฮาร์ดแวร์การ์ดชนิด Micro SD

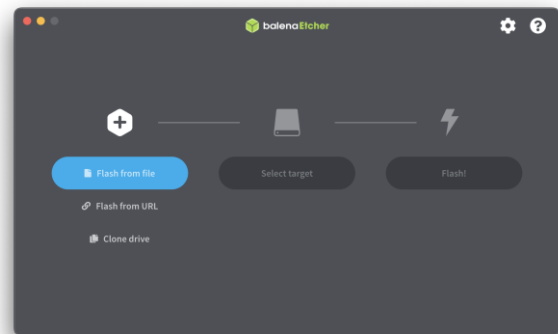
ฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์ มีระบบปฏิบัติการทำงานหลักและรอง ซึ่งแยกเฟิร์มแวร์สำหรับระบบปฏิบัติการและแอปพลิเคชันรันไทม์ ซึ่งทำให้ผู้พัฒนาสามารถอัปเดตพัฒนาโปรแกรมบนแอปพลิเคชันบนระบบปฏิบัติการรองที่ติดตั้งบนการ์ดชนิด SD โดยผู้พัฒนาไม่จำเป็นต้องเข้าไปสัมผัสระบบปฏิบัติการหลัก คือ หน่วยความจำ QSPI ที่อยู่ใน SOM ซึ่งตั้งโปรแกรมไว้ล่วงหน้า (อิมเมจ QSPI ที่โหลดไว้ล่วงหน้า) จากโรงงาน ระบบปฏิบัติการรอง คืออินเทอร์เฟซบนการ์ดชนิด SD

ขั้นตอนสำหรับการติดตั้งระบบปฏิบัติการบนการ์ดชนิด SD สามารถทำได้ตามขั้นตอนดังนี้

4.1.1. ดาวน์โหลด Kria™ KR260 Robotics Starter Kit Image จากเว็บไซต์ และบันทึกไว้ในคอมพิวเตอร์

<http://ubuntu.com/download/xilinx>

4.1.2. จากรูปที่ 6 ดาวน์โหลดโปรแกรม Balena ใช้ได้กับ Window, Linux และ macOS



รูปที่ 6 โปรแกรม Balena สำหรับดาวน์โหลดไฟล์อิมเมจลงบนการ์ดชนิด Micro SD

4.1.3 เลือกไฟล์อิมเมจที่ดาวน์โหลด เพื่อแฟลชลงเมมโมรี่การ์ดชนิด Micro SD

4.2 ขั้นตอนการเชื่อมต่อสายสัญญาณเข้ากับฮาร์ดแวร์ประมวลผล Kria KR260

4.2.1 ติดตั้งเมมโมรี่การ์ดชนิด Micro SD ในช่องเสียบการ์ด microSD (J11) ดังรูปที่ 7

4.2.2 เชื่อมต่อสาย USB-A to micro-B ซึ่งรองรับการถ่ายโอนข้อมูล โดยเชื่อมต่อปลาย micro-B เข้ากับ J4 บนฮาร์ดแวร์ประมวลผล Kria KR260

4.2.3 เชื่อมต่อแป้นพิมพ์/เมาส์ USB เข้ากับพอร์ต USB (U44 และ U46)

4.2.4 เชื่อมต่อกับจอภาพ/จอแสดงผลโดยใช้สาย DisplayPort หรือ ตัวแปลง DisplayPort to HDMI ชนิด Active เท่านั้น

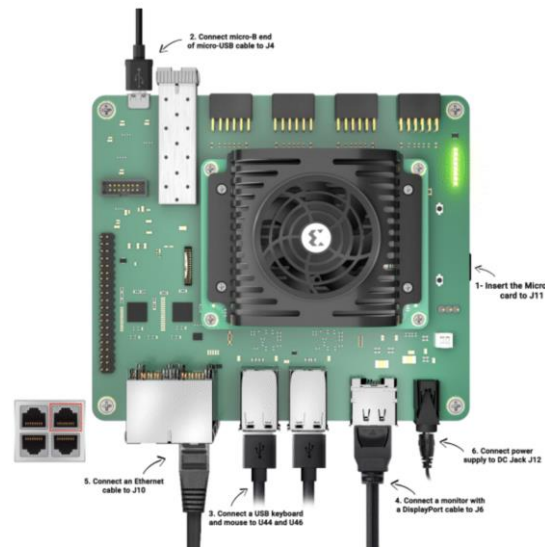
4.2.5 เชื่อมต่อสาย Ethernet เข้ากับพอร์ต PS ETH (J10D) ซึ่งเป็นพอร์ตบนขา) สำหรับการเข้าถึงอินเทอร์เน็ต

4.2.6 ติดตั้งแหล่งจ่ายไฟและเชื่อมต่อกับแจ็ค DC (J12) บนฮาร์ดแวร์ประมวลผล Kria KR260

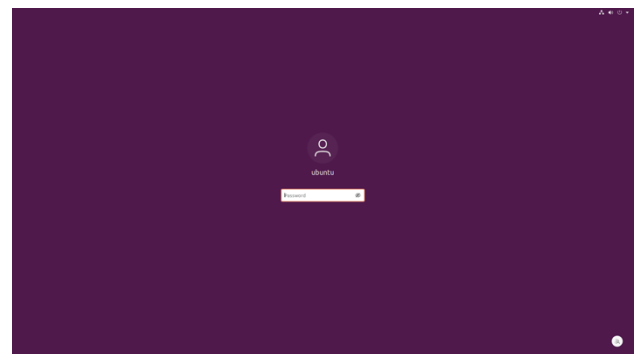
4.2.7 สามารถเชื่อมต่อกล้อง/เว็บแคม USB ที่รองรับกับฮาร์ดแวร์ประมวลผล Kria KR260

4.3 การติดตั้งโปรแกรมเพิ่มเติมบนฮาร์ดแวร์ประมวลผล Kria KR260

เมื่อผู้พัฒนาต้องการใช้งานระบบปฏิบัติการลินุกซ์ฮาร์ดแวร์ประมวลผล Kria KR260 ต้องมีการเชื่อมต่อแป้นพิมพ์ เมาส์ และจอภาพ เพื่อให้ควบคุมได้ง่ายขึ้น มีขั้นตอนสำหรับการติดตั้งและใช้งานดังนี้ ต่อแหล่งจ่ายไฟเข้ากับปลั๊กไฟ ไฟแสดงผลแอลอีดีควรติดสว่าง และหลังจากนั้นประมาณ 10-15 วินาที เอาต์พุตของคอนโซลบนจอแสดงผล หลังจากผ่านไปประมาณหนึ่งนาที หน้าจอการเข้าสู่ระบบ ควรปรากฏตามรูปที่ 10 โดยที่ฮาร์ดแวร์ประมวลผล Kria KR260 จะทำงานทันทีเมื่อเสียบปลั๊กไฟ ซึ่ง Kria KR260 จะไม่มีสวิตช์เปิด/ปิดบนบอร์ด



รูปที่ 7 การเชื่อมต่อสายสัญญาณเข้ากับฮาร์ดแวร์ประมวลผล Kria KR260 [6]

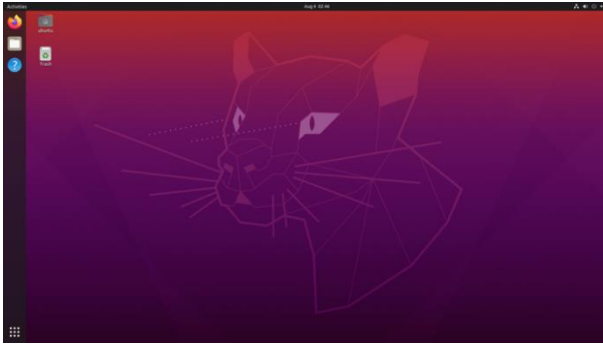


รูปที่ 8 แสดงผลหน้าจอการเข้าสู่ระบบ

จากรูปที่ 8 การเข้าใช้งานระบบโดยมีชื่อผู้ใช้และรหัสผ่านดังเดิมดังนี้

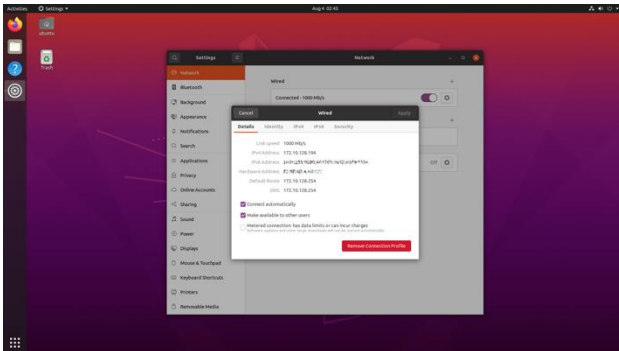
username: ubuntu

password: ubuntu



รูปที่ 9 หน้าจอระบบปฏิบัติการลินุกซ์ข้อบังคับ

จากรูปที่ 9 เมื่อเข้าสู่ระบบแล้วผู้ใช้งานจะเห็นหน้าเดสก์ท็อปของระบบปฏิบัติการลินุกซ์ข้อบังคับ เวอร์ชัน 22.04 (Ubuntu 22.04) ดังรูปที่ 11



รูปที่ 10 เมื่อเชื่อมต่ออินเทอร์เน็ตผ่านสาย RJ45 จะได้รับ IP จากเราเตอร์

เพื่อทดสอบการใช้งานอินเทอร์เน็ตรูปที่ 12 เปิดเทอร์มินัลและตรวจสอบการเชื่อมต่ออินเทอร์เน็ตผ่าน "ping" หรือ "DNS lookup"



สังเกตว่าการส่ง/รับแพ็กเก็ตทำงาน และไม่มีการสูญเสียแพ็กเก็ตด้วยคำสั่ง ping ด้านบน แสดงว่าการเชื่อมต่ออินเทอร์เน็ตใช้งานได้

ติดตั้ง xlnx-config snap ที่จำเป็นสำหรับการจัดการระบบ โดยมีวัตถุประสงค์หลักของ xlnx-config คือ การโหลดแพลตฟอร์มฮาร์ดแวร์แบบกำหนดเอง โดยเป็นส่วนหนึ่งของไบรบบรองบน

ระบบปฏิบัติการข้อบังคับสำหรับฮาร์ดแวร์ประมวลผล Kria KR260 ของบริษัท Xilinx

```
sudo snap install xlnx-config --classic --channel=2.x
```

สำหรับข้อมูลเพิ่มเติมของการใช้งาน xlnx-config snap ได้จาก

<https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/2057043969/Snaps+-+xlnx-config+Snap+for+Certified+Ubuntu+on+Xilinx+Devices>

เมื่อเรียกใช้คำสั่ง sysinit xlnx-config เพื่อติดตั้ง Gstreamer เวอร์ชัน Xilinx แบบกำหนดเอง เมื่อมีการแจ้งเตือนให้ยืนยันค่าเริ่มต้นทั้งหมด

```
xlnx-config.sysinit
```

หากผู้พัฒนาต้องการการเปิดใช้แอป Kria บางตัว จำเป็นต้องใช้พอร์ตอนุกรม USB-UART เพื่อเข้าถึงอินเทอร์เฟซบรรทัดคำสั่งแทนที่จะใช้เดสก์ท็อป

5. แอปพลิเคชันบนฮาร์ดแวร์ประมวลผล Kria KR260 สำหรับหุ่นยนต์

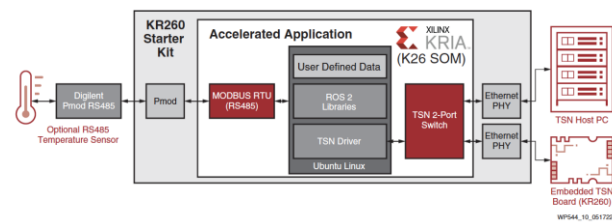
แนวคิดของแอปพลิเคชันที่ส่งผ่าน Xilinx App Store มีความสำคัญอย่างยิ่งในการนำประโยชน์ของ Xilinx มาสู่ผู้ใช้ใหม่ที่ไม่มีความรู้ในด้านการออกแบบ FPGA บริษัท Xilinx และพันธมิตรกำลังสร้างแอปพลิเคชันที่เร่งความเร็วขึ้นเพิ่มเติม ในแง่หนึ่งคือการออกแบบอ้างอิงระดับการผลิต แอปพลิเคชันเร่งความเร็วที่สร้างไว้ล่วงหน้าแต่ละรายการจะแปลง K26 SOM เป็นกล่อง 10

GigE Vision หรือโหนดการสื่อสาร ROS 2 TSN หรือโหนดการรับรู้ ROS 2 เป็นต้น แต่ละแอปพลิเคชันใช้ประโยชน์จากฮาร์ดแวร์ K26 ที่ทนทานแบบเดียวกับกลไกคอมพิวเตอร์ แต่ละแอปพลิเคชัน ปรับใช้ตัวเร่งความเร็วของฮาร์ดแวร์ที่แตกต่างกันซึ่งสร้างขึ้นตามวัตถุประสงค์โดยเฉพาะสำหรับการกำหนดเป้าหมายแอปพลิเคชันปลายทาง

ROS 2 Perception Node: แอปพลิเคชันนี้เปิดใช้งานการรับรู้ของหุ่นยนต์ที่มีประสิทธิภาพสูงผ่านการเร่งฮาร์ดแวร์ของส่วนประกอบของสแต็คการรับรู้ ROS 2 เปิดใช้งานการไหลของการจำลอง Gazebo และเป็นเส้นทางสู่การออกแบบด้วย KRS

5.1 ตัวอย่างการออกแบบ ROS 2 Multi-Node Communications ผ่าน TSN

สร้างโหนดผู้ส่งและผู้รับ ROS 2 DDS โดยใช้การสื่อสาร TSN MAC-layer ผ่านการเชื่อมต่ออีเธอร์เน็ต ดูรูปที่ 6

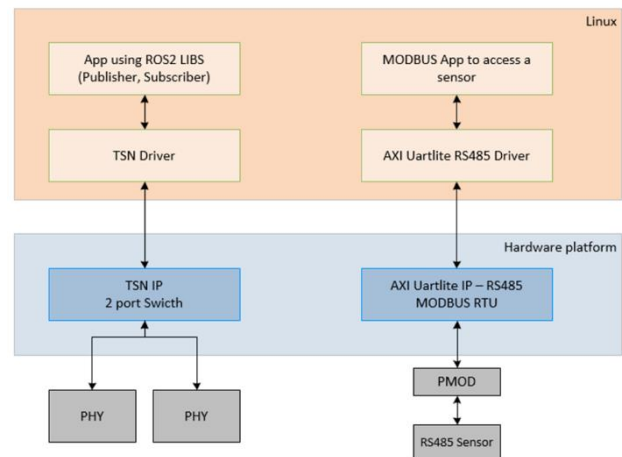


รูปที่ 11 ROS 2 Multi-Node Communications via TSN [7]

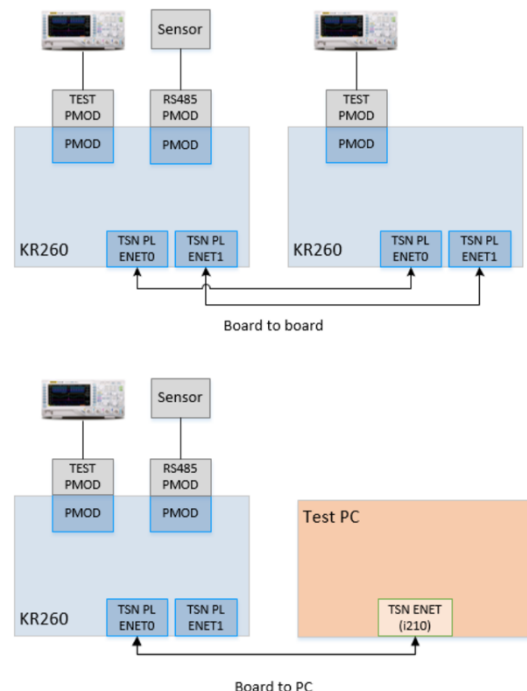
ในแอปพลิเคชันนี้ ฟังก์ชันตัวส่งและตัวรับจะอยู่บนบอร์ดสองบอร์ดแยกกันและสื่อสารผ่านอีเธอร์เน็ต PHY TSN IP ในแพลตฟอร์มฮาร์ดแวร์รองรับการซิงโครไนซ์ทางเวลาระหว่างอุปกรณ์ นอกจากนี้ยังอนุญาตให้สร้างกราฟฟิคตามคลาสกราฟฟิค AXI Uartlite IP ช่วยให้สามารถสื่อสารกับอุปกรณ์ RS485 ผ่านโปรโตคอล MODBUS RTU

ระบบปฏิบัติการหุ่นยนต์ (ROS) เป็นชุดไลบรารีซอฟต์แวร์และเครื่องมือมากมายสำหรับการสร้างแอปพลิเคชันหุ่นยนต์ นอกเหนือจากสิ่งที่เป็นนามธรรมสำหรับอัลกอริทึมที่ใช้กันทั่วไป รูปแบบการส่งข้อความ กรอบงานสร้าง มายังอนุญาตและมีตัวเลือกสำหรับมิดเดิลแวร์ ROS 2 สร้างขึ้นบน DDS/RTS เป็นมิดเดิลแวร์ ซึ่งให้ความสามารถในการปรับขนาดได้เนื่องจากแนวทางของผู้

เผยแพร่-ผู้ใช้บริการเป็นศูนย์กลางข้อมูล แอปพลิเคชันนี้ใช้การใช้งาน DDS ของมิดเดิลแวร์เริ่มต้นสำหรับการส่งข้อความระหว่างโหนดต่าง ๆ ผ่านเครือข่าย TSN



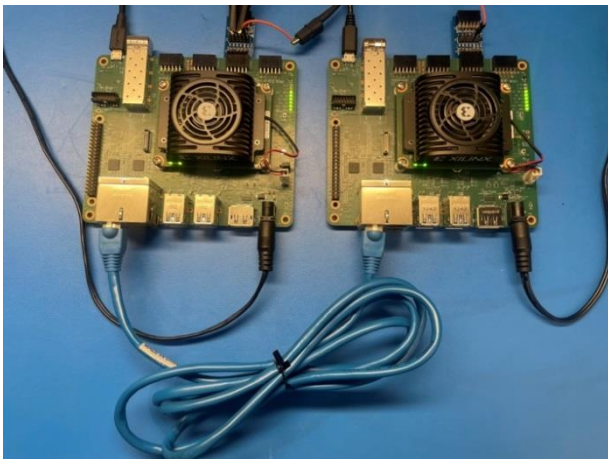
รูปที่ 12 การออกแบบ ROS 2 Multi-Node Communications via TSN [7]



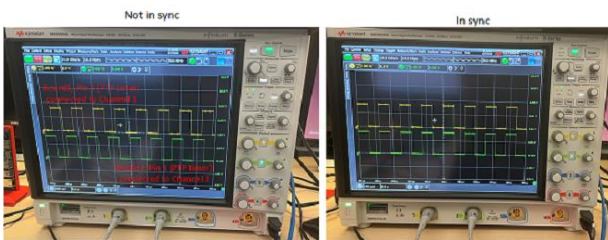
รูปที่ 13 แสดงแอปพลิเคชันนี้รองรับ Topology สองแบบ

จากรูปที่ 13 แอปพลิเคชันนี้รองรับ Topology สองแบบ คือ
1) การเชื่อมต่อระหว่างฮาร์ดแวร์ประมวลผล Kria KR260 กับ
ฮาร์ดแวร์ประมวลผล Kria KR260 ผ่านพอร์ต RJ45 2) การเชื่อมต่อ
ฮาร์ดแวร์ประมวลผล Kria KR260 กับ คอมพิวเตอร์ พอร์ต RJ45

โดยผู้พัฒนาสามารถใช้ออสซิลโลสโคปเพื่อตรวจสอบการ
ซิงโครไนซ์นาฬิกา จากรูปที่ 14 สามารถตรวจสอบรูปแบบ
การจราจรด้วยสัญญาณที่ส่งไปยัง Test PMOD แบบ 12 พิน



รูปที่ 14 การตั้งค่าและเชื่อมต่อฮาร์ดแวร์ประมวลผล Kria KR260
ในรูปแบบที่ 1 [8]

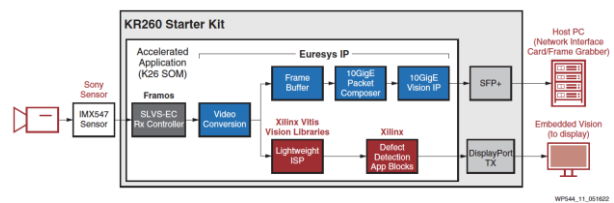


รูปที่ 15 การวัดสัญญาณด้วยออสซิลโลสโคป ผ่าน PMOD [8]

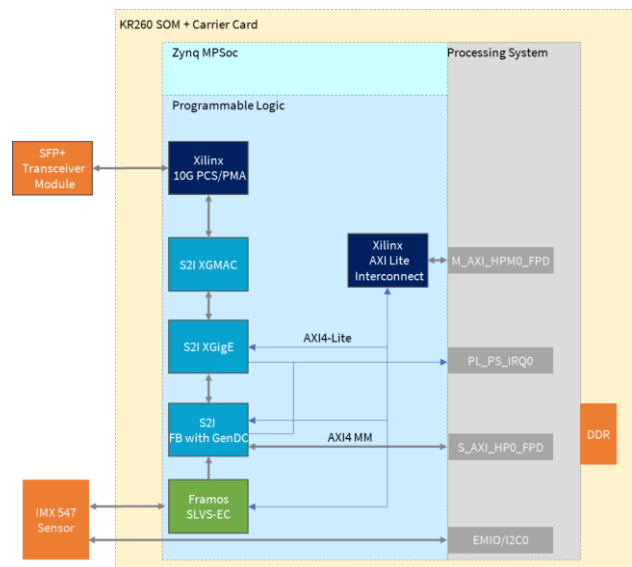
เมื่อใช้ออสซิลโลสโคป รูปที่ 15 หรือ Analog Discovery 2
และแอปพลิเคชัน WaveForms นาฬิกาจะเริ่มไม่ซิงค์กันและจะเริ่ม
ซิงค์กัน ซึ่งจะสังเกตได้จากการจัดตำแหน่งเฟสระหว่าง Channel 1
และ Channel 2

5.2 ตัวอย่างการออกแบบ 10GigE Vision Camera

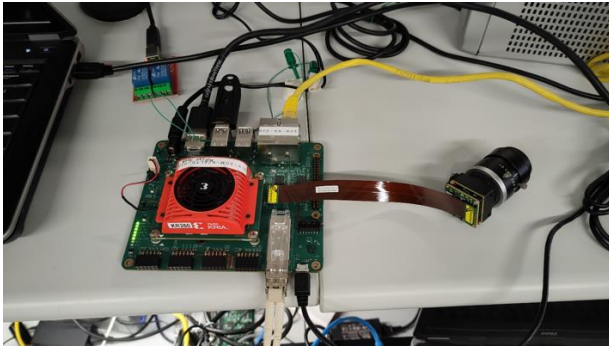
แอปพลิเคชันในกล้องอัตโนมัติในโรงงานเดียวโดยใช้อินพุต
เซ็นเซอร์ SLVS-EC และกล้อง GigE Vision ไปยังเอาต์พุต PC
พร้อมกล้องวิชั่นเพิ่มเติมโดยตรงไปยังจอภาพที่แสดงอัลกอริทึม
OpenCV การตรวจจับข้อบกพร่องในตัว เซ็นเซอร์ภาพขาวดำหรือ
เซ็นเซอร์ภาพสี Sony IMX547 ที่ใช้งานร่วมกันได้มีจำหน่ายเป็น
อุปกรณ์เสริมโดยมีค่าใช้จ่ายเพิ่มเติม ดังรูปที่ 16 สำหรับการ
ออกแบบที่ใช้ใน Programmable Logic (PL) รูปที่ 17 ต่อไปนี้
แสดงสถาปัตยกรรมฮาร์ดแวร์ระดับบนสุดของการออกแบบ



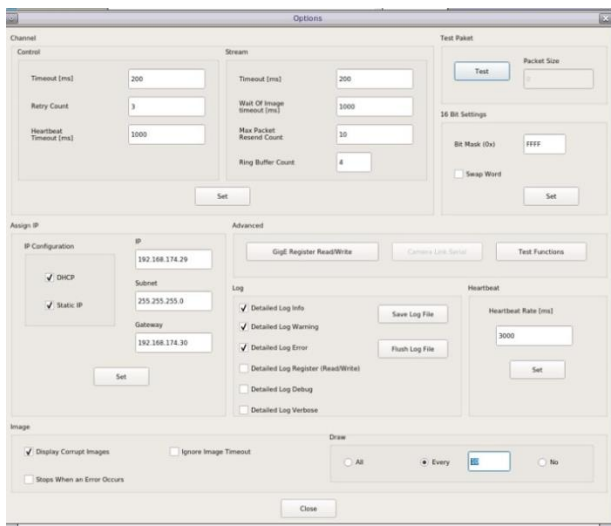
รูป 16 10GigE Vision Camera [7]



รูปที่ 17 การออกแบบระดับสูง ใช้ไปป์ไลน์ [8]



รูป 18 การเชื่อมต่อกล้อง SONY IMX547 กับฮาร์ดแวร์ประมวลผล Kria KR260 [9]



รูปที่ 19 โปรแกรม sphinx GEV viewer

โปรแกรม sphinx GEV viewer รูปที่ 19 สามารถเปลี่ยนค่าได้จาก host บนระบบปฏิบัติการลินุกซ์และวินโดวส์

โดยแอปพลิเคชันทั้งสองเป็นตัวอย่างการพัฒนาโปรแกรมบนฮาร์ดแวร์ประมวลผล Kria KR260 ที่มีความสามารถหลากหลายและสามารถประมวลผลได้ค่อนข้างรวดเร็วและมีประสิทธิภาพสูง

6. สรุป

ฮาร์ดแวร์ประมวลผล Kria KR260 Robotics Starter Kit เป็นแพลตฟอร์มเพื่อการพัฒนาหุ่นยนต์ สำหรับนักวิทยาการหุ่นยนต์ระดับเริ่มต้นและรวมไปถึงผู้มีประสบการณ์ที่ต้องการสร้างแนวคิดที่เป็นนวัตกรรมใหม่สำหรับวิทยาการหุ่นยนต์ที่ใช้ ROS 2 และนำไปใช้

ในการผลิต ชุดอุปกรณ์เริ่มต้นนี้มีอินเทอร์เฟซหลักสำหรับการพัฒนาแอปพลิเคชันสำหรับหุ่นยนต์ วิชันซิสเต็ม และการสื่อสารและการควบคุมทางอุตสาหกรรม แอปพลิเคชันแบบเร่งที่มีอยู่ใน Xilinx App Store และ Kria Robotics Stack (KRS) ทำให้ชุดเริ่มต้นเป็นแพลตฟอร์มที่พร้อมใช้งานได้ทันที เส้นทางสู่การปรับใช้การผลิตที่ราบรื่นผ่าน Kria SOM ช่วยให้นักพัฒนาโดยเฉพาะอย่างยิ่งในองค์กรขนาดเล็กและขนาดกลาง เข้าสู่ตลาดได้เร็วขึ้น ลดต้นทุนรวมในการเป็นเจ้าของ และเข้าถึงประโยชน์ FPGA พื้นฐานจากเวลาแฝงต่ำและการกำหนดระดับผ่าน Xilinx adaptive SoCs

เอกสารอ้างอิง

- [1] Advanced Micro Devices, Inc., Adaptive Computing in Robotics. [ออนไลน์]: <https://www.xilinx.com/products/som/adaptive-computing-in-robotics-ebook.html> [สืบค้น เมื่อ 3 พ.ย. 2565].
- [2] Frankfurt, World Robotics Report: "All-Time High" with Half a Million Robots Installed in one Year. [ออนไลน์]: <https://ifr.org/ifr-press-releases/news/wr-report-all-time-high-with-half-a-million-robots-installed> [สืบค้น เมื่อ 3 พ.ย. 2565].
- [3] S. Prongnuch and S. Sitjongsataporn, "Differential Drive Analysis of Spherical Magnetic Robot Using Multi-Single Board Computer," International Journal of Intelligent Engineering and Systems, Vol.14, No.4, pp. 264-275, 2021.
- [4] S. Prongnuch and S. Sitjongsataporn, "Thai Voice-Controlled Analysis for Car Parking Assistance in System-on-Chip Architecture," Advances in Technology Innovation, Vol.5, No.4, pp. 203-220, 2020.
- [5] S. Prongnuch and S. Sitjongsataporn, "Exterior Car Parking Assistance Algorithm Based on Reconfigurable System for Future Industry," Journal of Mobile Multimedia (JMM), Vol.16, No.1-2, pp. 203-220, 2020.
- [6] Karan Kantharia, Chetan Khona "Kria KR260 Robotics Starter Kit: Unleashing Roboticists through Hardware Acceleration" [ออนไลน์]: https://china.xilinx.com/content/dam/xilinx/support/documents/white_papers/wp544-kr260.pdf [สืบค้น เมื่อ 3 พ.ย. 2565].
- [7] Víctor Mayoral-Vilches, "Kria Robotics Stack A ROS 2-centric Approach for Hardware Acceleration in Robotics" [ออนไลน์]: https://xilinx.eetrend.com/files/2021-09/wen_zhang_/100553889-221299-wp540-kria-robotics-stack.pdf [สืบค้น เมื่อ 3 พ.ย. 2565].
- [8] "ROS 2 Multi-Node Communications via Time-Sensitive Networking (TSN)" [ออนไลน์]: https://xilinx.github.io/kria-apps-docs/kr260/build/html/docs/ros2_multinode_communication_via_tsn/ros2_multinode_communication_via_tsn_landing.html [สืบค้น เมื่อ 30 พ.ย. 2565].
- [9] "The Machine Vision (MV) camera application" [ออนไลน์]: https://xilinx.github.io/kria-apps-docs/kr260/build/html/docs/10gige_vision_camera/docs/overview.html [สืบค้น เมื่อ 30 พ.ย. 2565].



อภิรักษ์ อธิณณมิต จบการศึกษาคณะวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า (สาขาย่อยโทรคมนาคม)มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ และวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า (สาขาย่อยโทรคมนาคม) มหาวิทยาลัยเทคโนโลยีมหานคร ปัจจุบันเป็นอาจารย์ประจำ

สาขาวิชาวิศวกรรมหุ่นยนต์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสวนสุนันทา งานวิจัยที่สนใจ ได้แก่ การออกแบบวงจรอิเล็กทรอนิกส์ อากาศยานไร้คนขับ และการสื่อสารไร้สาย



ปญญานันท์ แสงอรุณ จบการศึกษาศิลปศาสตรบัณฑิต สาขาการออกแบบบรรจุภัณฑ์ มหาวิทยาลัยมหาวิทยาลัยรัตนเรศวร และ วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีการออกแบบผลิตภัณฑ์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร

ลาดกระบัง ปัจจุบันเป็นหัวหน้าสาขาการออกแบบผลิตภัณฑ์และบรรจุภัณฑ์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสวนสุนันทา งานวิจัยที่สนใจ ได้แก่ การออกแบบผลิตภัณฑ์และบรรจุภัณฑ์ การพัฒนาการออกแบบผลิตภัณฑ์และบรรจุภัณฑ์เพื่อชุมชน บรรจุภัณฑ์ที่สืบสานวัฒนธรรม



เศรษฐกาล โปร่งนุช จบการศึกษาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร และวิศวกรรมศาสตรมหาบัณฑิต กับวิศวกรรมศาสตรดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า สาขาย่อยวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีมหานคร ปัจจุบันเป็นหัวหน้าสาขาวิชาวิศวกรรม

หุ่นยนต์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสวนสุนันทา ดำรงตำแหน่งผู้ช่วยศาสตราจารย์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ งานวิจัยที่สนใจ ได้แก่ ระบบคอมพิวเตอร์และสถาปัตยกรรมคอมพิวเตอร์ ระบบสมองกลฝังตัว เอฟพีจีเอ ระบบที่มีตัวประมวลผลหลายชนิด และหุ่นยนต์